(1)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DOWNSIZING INFORMATION SYSTEMS:
FRAMING THE ISSUES FOR
THE OFFICE OF NAVAL INTELLIGENCE (ONI)

by

Peter M. Hutson
March 1994

Thesis Co-Advisors: James C. Emery
Myung W. Suh

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
March 1994

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
DOWNSIZING INFORMATION SYSTEMS:
FRAMING THE ISSUES FOR THE OFFICE OF NAVAL INTELLIGENCE (ONI)

5. FUNDING NUMBERS

6. AUTHOR(S)
Peter M. Hutson, Lieutenant, USN

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93940

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Mr. Brent Snyder, Head Telecommunications Management Department
Office of Naval Intelligence (ONI) – 62
4600 Silver Hill Road, Washington, D.C. 20389-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)
Downsizing information systems from large and centralized mainframe computing architectures to smaller and distributed desktop systems is one of the most difficult and critical strategic decisions facing both corporate and government organizations. Vendor advertisements and media hype often boast of huge cost savings and greater flexibility while retaining mainframe-strength performance. Cryptic terminology, biased vendor assistance, and rapidly changing technology complicate already difficult decisions. This thesis provides an executive summary for middle and top managers requiring a survey of the major downsizing issues. It provides an overview of architectural trends that are helping to fuel the downsizing process to include an emphasis on the client/server paradigm, the evolving roles of the mainframe and desktop computers, and innovative architectural software tools. An analysis of management and technical risks according to organizational, performance, and cost factors also focuses on such critical considerations as business process reengineering, open systems, flexibility, throughput, security, conversion costs, and life-cycle costs of downsized systems. Finally, this thesis "frames" the issues for the Office of Naval Intelligence and DoD by highlighting and outlining some general guidelines that may be used to intelligently plan strategy and make key organizational decisions.

14. SUBJECT TERMS
Downsizing, Information Systems, Client/Server, Distributed Computing, Computing Architectures

15. NUMBER OF PAGES 115

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 DTIC QUALITY INSPECTED 2 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102

Approved for public release; distribution is unlimited.

DOWNSIZING INFORMATION SYSTEMS: FRAMING THE ISSUES FOR THE OFFICE OF NAVAL INTELLIGENCE (ONI)

by

Peter M. Hutson

Lieutenant, United States Navy
B.S., United States Naval Academy, 1987

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1994

Author: Peter M. Hutson
Peter M. Hutson

Approved by: M. Suh
Dr. Myung W. Suh, Thesis Co-Advisor

James C. Emery
Dr. James C. Emery, Thesis Co-Advisor

Reuben T. Harris for
Dr. David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

Downsizing information systems from large and centralized mainframe computing architectures to smaller and distributed desktop systems is one of the most difficult and critical strategic decisions facing both corporate and government organizations. Vendor advertisements and media hype often boast of huge cost savings and greater flexibility while retaining mainframe-strength performance. Cryptic terminology, biased vendor assistance, and rapidly changing technology complicate already difficult decisions. This thesis provides an executive summary for middle and top managers requiring a survey of the major downsizing issues. It provides an overview of architectural trends that are helping to fuel the downsizing process to include an emphasis on the client/server paradigm, the evolving roles of the mainframe and desktop computers, and innovative architectural software tools. An analysis of management and technical risks according to organizational, performance, and cost factors also focuses on such critical considerations as business process reengineering, open systems, flexibility, throughput, security, conversion costs, and life-cycle costs of downsized systems. Finally, this thesis "frames" the issues for the Office of Naval Intelligence and DoD by highlighting and outlining some general guidelines that may be used to intelligently plan strategy and make key organizational decisions.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank the Mr. Brent Snyder, CDR Douglas Beauvais, and Mr. Donald Gerbozy of the Office of Naval Intelligence for sponsoring this research topic and for providing the support necessary for its completion. I would also like to express my appreciation to Professors James Emery and Professor Myung Suh for their insights, guidance, and assistance during the development and refinement of this thesis. Finally, I certainly cannot omit mention of my wife, Jeanne, who has been a critical and constant source of moral support throughout this whole process.

I. INTRODUCTION

A. PURPOSE

The purpose of this thesis is to provide mid-level and top-level managers at the Office of Naval Intelligence (ONI) with a risk assessment and "state of the art" report on factors associated with, and affecting, the key decision to "downsize" information systems. Most often typified by the architectural transition from large, centralized mainframe systems to a network of distributed workstations, the downsizing issue is currently one of the most difficult and critical strategic decisions facing both corporate and government organizations. The decision to downsize is particularly important in light of the new budget constraints, the reduction of personnel associated with the tighter budget, and resulting pressure to accomplish "more with less." Few would dispute the harkening of a new world era— the post-industrial *age of information* —where efficient collection, processing, and dissemination of *information* is becoming a strategic necessity. This thesis is written as a consequence of new technologies that have enabled *better, faster, and cheaper* options of processing that information. These new innovations in technology have generated new methods of processing information, new *architectures*. Those organizations that understand when, where, and how to select the most advantageous of these architectures will better be able to meet their budget, achieve maximum productivity from their personnel, and accomplish defined goals. This thesis will try to provide some background, insight, and suggestions to the "where, when, and how" of downsizing.

Though this thesis is written in response to research requirements prompted by mid-level and top-level managers at ONI, significant portions this thesis are applicable to any large-scale organization, government or corporate, with sizable investments in information systems. The research conducted in the completion of this thesis primarily dealt with literature reviews and case studies associated with private business organizations. Undeniably, however, there are lessons to be learned by DoD through the study of the corporate world. After examination of the most widely relevant downsizing issues, the final segment of this thesis will briefly "frame" the most pertinent issues for DoD organizations in general, and ONI in particular.

B. BACKGROUND

This thesis is being written for the Office of Naval Intelligence (ONI) located in Suitland, Maryland. ONI is responsible for the collection, production, and dissemination of Naval-related intelligence information to satisfy the requirements of the Department of the Navy (DoN), operating forces and commands, the research and development (R&D) community, the Department of Defense (DoD), the Defense Intelligence Agency (DIA), and national command authorities and agencies.

In support of its mission, ONI manages a large array of naval intelligence automated data processing systems that includes older technology mainframe computer systems. These computer systems serve as hosts for extremely large databases supporting mission areas such as scientific and technical data, worldwide merchant shipping, counternarcotics, digital imagery, inter alia.

The current and future fiscal environment mandates that ONI initiate cost savings programs within the automated systems department, while ensuring these efforts do not adversely impact the operational support mission of the command. These reductions include cuts in system support hardware, software life-cycle support, and training costs. This situation is being exacerbated with increasing requirements such as new on-line databases as part of the worldwide Department of Defense Intelligence Information System (DODIIS) Wide Area Network (WAN). Finally, in mid-1994, ONI will be moving into a new building—currently under construction—and wants to examine the issues of replacing large mainframes with smaller workstation systems in the new environment without losing functionality.

ONI submitted a proposal for a thesis research topic to the Naval Postgraduate School (NPS) that suggested a risk analysis "on downsizing from large mainframe computer systems to high-end, client-server workstation as it applies to system performance, personnel reductions, and system cost savings." One of ONI's proposed approaches was for NPS to perform a systems-level risk analysis based upon a case study of a major commercial firm, such as Mobil Oil or the IBM Credit Union, which downsized from larger mainframe computer systems to high-end, client/server workstations. ONI made it clear that a case study was only a suggested approach and left it to researcher discretion to determine the most appropriate manner to address the pertinent downsizing issues. As already hinted in the previous section and outlined in the next section, it was ultimately determined that an academic overview and survey of the most relevant downsizing issues would better suit ONI and DoD needs than a focus on one particular case study.

ONI's proposed research suggested focusing on lessons learned in the downsizing process including technical systems performance problems, cost savings, and management issues. ONI was also interested in how new technologies could be inserted into their information systems to help reduce the risk factors in downsizing the automated data processing systems, while simultaneously optimizing system functionality, manpower reduction tradeoffs, and system hardware and software maintenance costs.

A visit was made to ONI to ascertain the scope of the downsizing envisioned and to enable tailoring of the research for use by the command. During the visit, the thesis sponsors emphasized that they would like the research to focus not on the command itself, but other corporate models. They did not want an "answer" on how ONI should, in detail, downsize. They believed that with an understanding and appreciation of ONI's size and complexity, appropriate case studies and an academic approach would best serve their purposes.

ONI's intention is to stay abreast of fast moving technology and to meet growing customer demands and requirements. The command is investigating new options, new technology, and new opportunities. The most appealing option that ONI is investigating is downsizing to a client/server environment with high-end performance workstations. If the command can increase mission effectiveness and save money through such an option then it will further pursue that objective.

C. THESIS SCOPE AND OUTLINE

The thrust of this thesis will be a survey of the trends, concepts, methodologies, and management strategies associated with the downsizing process. Because the issues related to downsizing involve both management and technological issues, each dependent on the other, this paper will address both management and technical perspectives.

The downsizing momentum has been fed by these two equally significant forces. On one hand, technology has been driving managers to re-think they way their organizations handle information. It has required top managers to streamline and re-engineer their business processes in order to capitalize on new technological opportunities. On the other hand, pressures to do "more with less" and subsequent implementations of new management schemes (e.g., Total Quality Leadership (TQL) and Business Process Reengineering (BPR)) have forced managers to consider how technology might assist them in coping with new management initiatives. The downsizing of the corporate structure is directly linked to the downsizing of information systems. As corporations aim to streamline their organization and personnel, they look to technology for solutions. Thus, neither the management or technical aspects of downsizing information systems should be considered without giving attention to the other; the two are inextricably linked. This thesis will reflect that relationship.

Before outlining the specific topics of discussion in this thesis, it is important to put a bound on this obviously broad topic in which many sub-topics, by themselves, could easily be addressed in separate theses. This thesis has an ambitious objective in that any of the subsequent chapters have been tackled by many different authors in many volumes of books and other publications. Here, I point out that this thesis will be a broad-brush survey of the downsizing issues, not unlike an executive "white paper" typical of larger organizations. This thesis will identify the major issues related to downsizing without delving into too much detail. It will provide a comprehensive analysis of both management and technical issues, putting them in context with trends in computing. Most importantly, this thesis will "frame" the downsizing trend, not offering any one solution, but offering ways of thinking about downsizing and methods of approaching it in the context of one's own organization.

Downsizing information systems can regarded as an inevitable result of evolving, more capable technology. Simply put, the evolution of computing technology from the mainframe in the "glass house" to the personal computer on the desktop has created a revolution. This "revolution in computing" has generated new ways of thinking about information technology as an integral part of an organization's strategy. The advent of desktop computing has enabled managers to think of new ways that IT can help an organization obtain its objectives. Top managers have been used to thinking about computing within a specific "mold" or context. Now, new technology has created an environment for a new way of thinking about computing.

The second chapter in this thesis (the first chapter being this introduction) encapsulates this new way of thinking, or "paradigm shift." This chapter first sets the stage by describing the changing business environment and the accompanying mandate for change. It describes how IT has become closely linked with strategy, and how IS architectures are tied to those strategies. Finally, it summarizes the shift from traditional mainframe computing to desktop computing and the resulting trend of downsizing.

The third chapter will provide an overview of architectural decisions related to the downsizing process. This will include a focus on establishing the appropriate architecture with emphasis on the client/server model, and the evolving role of the mainframe, mini-computer, and microcomputer within that architecture. The last part of this chapter will outline the various architectural tools and architectural trends that have helped fuel the downsizing movement and have helped change the way organizations view computing.

The fourth chapter in this thesis discusses risk analysis issues and factors that must be considered in the decision making process of whether or not to downsize. The chapter involves a discussion of both qualitative and quantitative issues. First, it is imperative to look at the organization as a whole and to define IT's role within it. IT-related processes need to be examined and evaluated in conjunction with the organizational goals. Here, a brief overview of some of the new management initiatives will be conducted. Next, this chapter shifts from an organizational aspect of risk analysis to an examination of performance factors. Again, in discussing performance-related considerations, the discussion will survey a wide gamut of issues from speed to security to maintenance of systems. Finally, in the last part of the chapter, I will discuss methods of cost-justifying downsizing. This will entail both quantitative and qualitative evaluations.

The final chapter will "frame" and summarize the key downsizing issues relative to DoD and ONI requirements. This will include a discussion of how the corporate environment downsizing considerations may differ from those of the military. Corporate "lessons learned" will be extracted and highlighted to give top managers some general guidelines from which they can intelligently plan strategy and make key decisions. Undoubtedly, the downsizing process has the potential to be an unstructured and haphazard exercise in frustration that can lead to huge cost over-runs and even IS failures. Hopefully, this final chapter will highlight the most applicable issues and provide a structured guideline to a complex and seemingly overwhelming process.

II. THE PARADIGM SHIFT

A. THE CHANGING ENVIRONMENT

Change is the only constant. This phrase is as applicable to the information systems world as it is to the current political and economic climate. Increased global competition, the need for global management, reduced product cycles, changing demographics, and the growing power of knowledge are all key forces driving escalating demands on information systems. The world is changing at a pace perhaps more rapid than ever before. What implications does this hold for informations systems level in both the Department of Defense and the corporate world? In this information age where knowledge means power, it translates into a need for information systems that are capable of handling huge volumes of data quickly, globally, securely, flexibly, and efficiently. In this post-industrial era, the new commodity is information. Those corporations, organizations, or even nations for that matter, that are capable of collecting, analyzing, and responding to new information will, in short, dominate the world stage.

B. INFORMATION TECHNOLOGY AS STRATEGIC ASSET

Presuming acceptance of the value and power of information, one readily sees the necessity of incorporating IT into a corporate or organizational strategy. IT is the engine of change that will propel some organizations into the future, while leaving others with less foresight to wither away in obsolescence. IT is the *enabling force* that permits the handling of collection, analysis, and intelligent responses to large volumes of information. Incremental differences in the capabilities and flexibility of those "engines" will clearly separate the winners from the losers. Thus, recognition of IT as a strategic asset and incorporating it as a core element into an organization's strategy is a first step in securing a foothold in this globally competitive market.

IT's strategic role can be categorized within three specific areas:

- as competitive tools
- to reengineer business processes
- for interorganizational linkage [Ref. 1:p. 69]

1. IT as a Competitive Tool and Strategic Necessity

Other factors being equal, information strategists readily recognize effective corporate implementation of IT as a competitive tool that is quickly becoming a strategic necessity. Organizations are realizing that if they fail to properly capitalize on this resource, they will simply not stay in business. William Davidow and Michael S. Malone, authors of *The Virtual Corporation*, acknowledge information processing as the "most important power source of our generation" in structuring and revitalizing the corporation for the 21st century [Ref. 2:p. 2]. At the corporate level, the authors state that "in the years to come, incremental differences in companies' abilities to acquire, distribute, store, analyze, and invoke actions based on information will determine the winners and losers in the battle for customers" [Ref. 2:p. 50].

In general, market dynamics do not permit organizations to maintain a competitive advantage with IT. Under unique circumstances, a competitive advantage can be sustained long enough that it can become a strategic advantage. Davidow and Malone provide such an example. Extending the argument beyond the corporate level to the military and illustrate, they illustrate the undeniable power of information in the Persian Gulf War:

> Why the rout? The answer was obvious to anyone watching television during the brief course of the war. The Allied troops had information on the Iraqis far superior to any information held by their enemy and then managed that information more effectively. In strategic command and control, the Allies used satellite and reconnaissance information to learn just about everything they needed to know about Iraq's static defenses. Tomahawk cruise missiles, using reconnaissance information stored in their memories, were so well programmed that they actually followed roads and

streets to their targets. Further, because the Allies appreciated the importance of information, they also knew the value of denying it to the other side.... With the Persian Gulf War, the value of information reached a zenith—bytes became more important than bullets. The case was made, with brutal clarity, that information could become the foundation for devastating destruction. [Ref. 2:p. 51]

Though this example may take the notion of strategic advantage to an extreme, it effectively illustrates the information systems role as a force multiplier for both business and defense applications and as a critical component in formulation of strategy.

2. Using IT to Reengineer Business Processes

Incorporation of IT into an organization's mission will not guarantee organizational success; technology, by itself, is not a panacea. A new IS may speed up an organizations's processes, but if the organization's processes are already broken, automation of those processes will only result in nominal improvements. IS managers should not rush out and assume that throwing technology at a problem will secure a competitive advantage.

Although the emphasis in the 1980s was on using IT for competitive advantage, the thrust in the early 1990s has turned inward, specifically to using IT as a catalyst to 'reengineer' outmoded business practices... mean(ing) fundamentally redesigning how the enterprise works—its procedures, control mechanisms, reporting relationships, decision makers, compensation criteria, and so forth—and generally making IT an integral part of operations. The goal is to rid the firm of ways of working that were appropriate for the paper-based world, and replace them with work modes that leverage the attributes of IT. [Ref. 1:p. 82]

Prior to utilization of any new architecture, the business processes must be examined. Business process re-engineering, according to IT consultant Michael Hammer—one of the gurus of business process reengineering—is about:

... scrapping every belief, every method, every process—and then rebuilding the organization in entirely new ways... Simply put, information technology is absolutely essential to re-engineering. It is the single most powerful tool for breaking traditional assumptions and rules, and the thing that makes new ways of operating possible. [Ref. 3:p. 11-12]

Recognition that competitive advantage can be gained through the use of technology is a preliminary step—but realization that a significant improvement in performance can be achieved only through an organization's reengineering of its business processes is essential. An organization needs to break down its functions, determine the value-added processes, and streamline the organization accordingly.

3. Using IT for Interorganizational Linkage

IT's third strategic role is inherent in its ability to provide automated communication mechanisms with other departments or organizations. With the globalization of the world economy, the need for interoganizational linkage is quickly growing. From the forging of strategic alliances to the maintenance of corporate accounting, the interchange of electronic data can allow organizations to maintain critical partnerships without the constraints of space and time. This flexibility is particularly significant for today's corporation that needs to be increasingly cognizant of and responsive to changing market trends and customer demands. The "virtual product" described in Davidow's and Malone's *virtual corporation* of the 21st century is the embodiment of the ideal:

> A virtual product (the term is used to mean both physical products and services) mostly exists even before it is produced. Its concept, design, and manufacture are stored in the minds of cooperating teams, in computers, and in flexible production lines... What these products and services have in common is that they deliver instant customer gratification in a cost-effective way. They frequently can be produced in diverse locations and offered in a great number of models or format... The ideal virtual product or service is one that is produced instantaneously and customized in response to consumer demand. [Ref. 2:p. 4]

Electronic Data Interchange (EDI), the business-to-business computer exchange of common business transactions, is one example of interorganizational linkage that has become pervasive throughout many different industries. The topic of interorganizational linkages is a thesis topic in itself; nevertheless, this aspect of IT undeniably warrants consideration as a critical component of an organizations's overall strategy.

C. THE CHANGING INFRASTRUCTURE

The previous section outlined three significant roles that IT can have in an organization's overall strategy. Now the goal of top managers must be to try define the IS architecture and its infrastructure that will enable their organizations to implement those strategies. The IS architecture is the means to the end. To a large degree, the molding of the IS architecture to the organizational strategy—thereby obtaining the "best fit"—will determine the degree of the strategy's success.

For large organizations dependent on mission-critical applications, the traditionally appropriate IS architecture has been the centralized architecture built around the capabilities of the mainframe. Until recently, for some corporations, the mainframe has been the only feasible solution to meeting performance requirements. The increased processing capability of the desktop computer and the advent of networking technology has increased the range of architectural options. The desktop revolution has unwrapped, to a degree, a mentality that has been fixated around the mainframe. This next section takes a look at these evolving performance trends—a discussion of traditional mainframe computing, the advance of desktop computing, and the resultant downsizing "paradigm shift."

1. Mainframe Computing

Mainframe computing has been the traditional computing architecture during the past three decades—the term *mainframe* being coined back in the early '60s with reference to the large and bulky metal cabinets where the central processing units (CPUs) were stored. The mainframe has essentially dominated the state of computing architectures since the development of the first digital computer (ENIAC) in the mid 1940s and the successful commercialization of the UNIVAC in the 1950s. It has been the "mold" that shaped the architectural paradigm for three decades. Companies have structured their processes around the mainframe, aware of both its overall strategic importance of computing power and the criticality of efficient information processing. Typically, the mainframes have hosted mission-critical applications, without which the organization could not continue to operate. Evolving from almost strictly scientific applications and batch processing to wide-ranging business functions and on-line transaction processing (OLTP), the mainframe has helped organizations collect and store increasingly voluminous amounts of data, process complex transactions quickly and effectively, and analyze and assimilate the resulting flood of data.

Though the mainframe has been characterized as the backbone of the computing industry and will surely continue to play a major role in the future of the computing industry, every major strength of the mainframe seems to inevitably result in a countering weakness. The impressive sophistication of mission-critical applications that meets all needs for all users, for example, is not only extremely expensive to develop, but typically frustratingly expensive and difficult to maintain. Compared to the new opportunities available on emerging alternatives, the significant strengths of the mainframe are becoming less attractive.

a. Major Shifts in Computing Requirements

The nature and role of today's mainframe has been set by a historical precedent. Until the 1970s, there was no real alternative to mainframe computing; large-scale computing was the only feasible solution to computing problems. Additionally, the hardware at the time was extremely expensive and software was relatively cheap. Almost all applications were developed to run in the batch-mode. On-line transaction processing was still in its infancy. The relative simplicity of core applications processed in the batch-mode translated into relatively high productivity and throughput and resulted in respectable cost savings.

Dan Trimmer, author of *Downsizing: Strategies for Success in the Modern Computer World*, suggests that the today's business requirements have changed since the 1970s. He further states that the design precedents that have typified mainframe computers have handicapped their ability to meet the new demands. He categorizes those changing business requirements by three major shifts:

(1) Transition to an Interactive, On-Line Environment. Batch-mode processing does not meet the needs and requirements of today's end-user. With the exception of back-ups and maintenance of files, on-line transaction processing has become the norm for many organizations.

(2) New Variety in the Applications Spectrum. Requirements for applications have become much more complex than before. In addition to the core applications, end-users are demanding error-handling and unanticipated decision support capabilities to serve management demands.

(3) Sophistication of End-users. End users have become much more computer literate in the last two decades and are demanding more of their systems. [Ref. 4:p. 35]

b. Costly and Complicated Hardware and Software

Concerns regarding cost are probably the primary motivating factors that stimulate criticisms about the mainframe computer. Not unrelated to the high costs are misgivings about the associated complexity of related software. One underlying reason—linked to its historical precedent—is that the mainframe has attempted to be all things to all people.

(1) Hardware Costs. Because of the complex designs and low-volume manufacturing (in a declining market) of mainframe computers, the high hardware costs are one of its most distinguishing features.

Table 1: 3090 PROCESSOR PURCHASE PRICES [Ref. 5]

Model Number	Purchase Price
340	$2,125,000
500	$4,100,000
580	$5,715,000
620	$8,050,000
720	$10,545,000
820	$16,025,000
900	$22,280,000

As shown by Table 1's purchase prices of IBM 3090 processors, the mainframe costs pale next to mini and microcomputers in a cost comparison—despite advancing technology and falling prices of central processors and solid state memoı,. Not only does the complexity of the hardware elevate these high prices of central processors to this $2 million to $22 million range (in early 1990 figures), but also the combined need for pre-sales support and extensive marketing.

(2) Hardware Complexity. Due to industry attempts to serve all needs of all customers combined with the establishment of early standards, the hardware on the mainframe is not only extremely expensive, but also very complex. Elaborate input/output systems and complicated processor and processor-to-memory relationships exacerbate the

situation. Author Dan Trimmer refers to this development as the "over-engineered mainframe" and uses the diagram illustrated in Figure 1, below, to highlight the point.

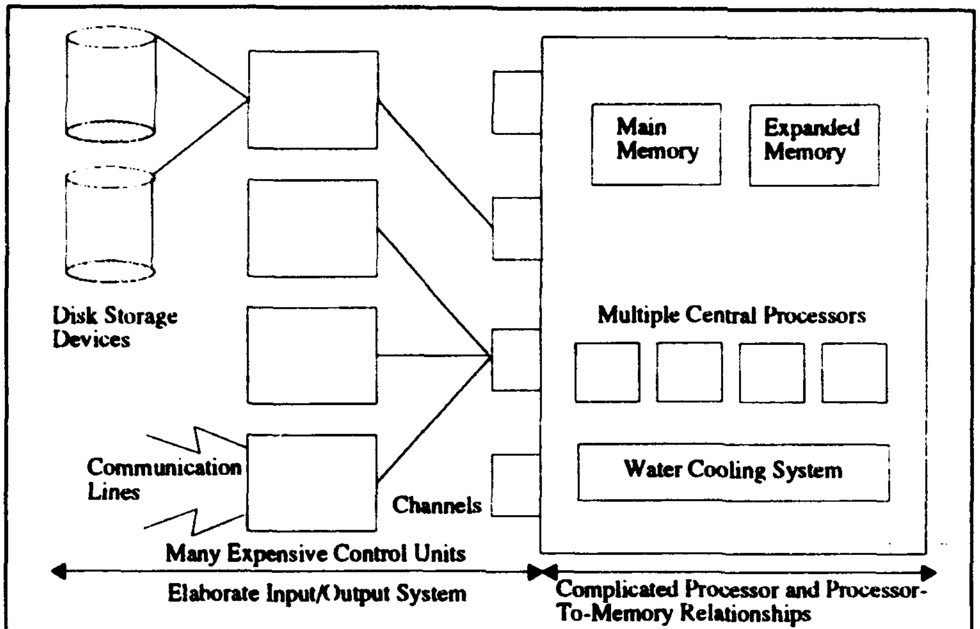


Figure 1:The "Over-engineered Mainframe" [Ref. 4:p. 39]

(3) *Software Costs.* Software packages developed for mainframes, like the mainframe hardware itself, tend to be very expensive. Unlike software for personal computers, mainframe software is developed by relatively few companies—resulting in less competition and higher prices. Furthermore, mainframe software developers simply are unable to distribute their high fixed costs of software development to the same large market share that the desktop computer enjoys. Those few corporations that purchase a common mainframe software package must absorb the high fixed costs among themselves (the variable costs of software development are relatively insignificant compared to the fixed costs). Also, because mainframe software is often custom developed for an organization, analysis of user requirements becomes critical and software must be tailored to meet specific needs.

This process is time consuming and costly. If requirements are not properly defined, however, the costs grow even larger.

In the personal computer market, the fixed costs of developing less sophisticated packaged software are simply not as high as those of the mainframe. Along with lower fixed costs and relatively insignificant variable costs, desktop software developers have the additional advantage of potentially exploiting a huge market base. This economic situation combined with the intense competition among personal computer software developers both serve to keep the price of desktop software very low.

(4) Software Complexity. The alphabet soup of software related to the core functionality of the various mainframes has proved to be detrimental to its continuing success. The common perception is that of incompatibility and proprietorship among the various systems. Table 2's outline of the various operating systems, transaction monitors, databases, and program generators common to large and small mainframes illustrates the problem.

Table 2: MAINFRAME SOFTWARE [Ref. 4:p.40]

Type of Machine	Transaction or Production	Information-Center
Large Mainframe	CSP	AS
	IMS-DB	DB2
	CICS	TSO
	MVS	MVS
Small Mainframe	CICS	CMS
	DL/1 DOS/VS	SQL/DS
	VSE	VM

As a result of this layering of diverse software among different types of mainframe systems, inevitable incompatibilities occur that ultimately affect functionality, performance, and costs. Attempts to overcome design shortfalls or compatibility problems often calls for introduction of conversion programs or software patches. Disregarding the maintenance nightmare that these supplementary "band-aids" cause, the mainframe software ultimately becomes corrupted by fragmentation—with a resulting higher overhead and greater inflexibility.

c. *The Widening Gap*

The three major shifts in the computing environment and the high costs and labyrinthine complexity of the mainframe computing environment are not the only perceived weaknesses of this fading monolith. As the mainframe remains plagued by its own problems, advances in the personal computer market has produced trends that the broad market has found quite appealing. Certainly, the demand has increased for "user-friendliness," graphical user interfaces (GUIs), object-oriented programming (OOP), fourth-generation language (4GLs), and open systems. The personal computer market, amid intense competition, has promptly responded to the call. Though the mainframe software developers have promised to deliver equivalent or better products in many cases, the evolution of those products has been too slow to suit the taste of the demanding user. Thus, as shown in Figure 2, the gap between the mainframe and popular alternatives grows wider.

The Gartner Group, a highly rated IS consulting firm, attributes the widening gap to the following:

- Traditional mainframes are too expensive. S/390 is the target for open systems, Unix, PCs and other challengers because it is expensive compared to alternatives and yet it still commands a significant share of the market.
- MVS is perceived to be proprietary and more closed than many other platforms.

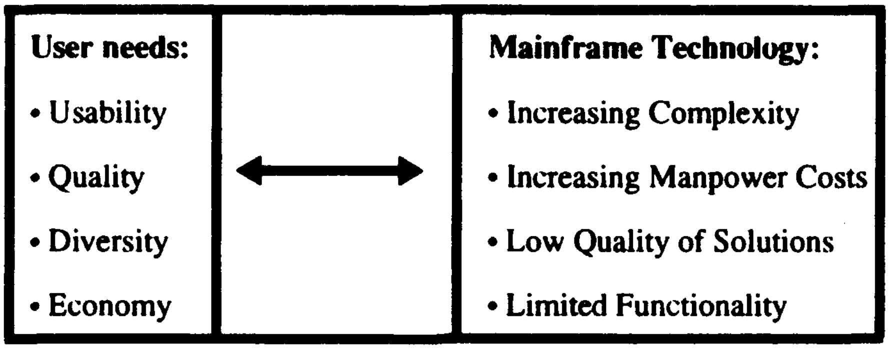


Figure 2: The Widening Gap [Ref. 4:p. 35]

- The pace of MVS software evolution has been slow. GUIs, object-oriented programming, network processing, including client/server, were deployed earlier on non-MVS platforms. [Ref. 7:p. 10-11]

d. The Mainframe Bureaucracy

Best typified by the image of the "white coated men behind glass doors," the traditional mainframe's presence is usually accompanied by excessive bureaucratic baggage that all too often exacerbates inefficiencies and wasteful administrative overhead. Unfortunately, with all of its inherent complexity in the software and hardware, this team of "experts" is not uncalled for. Organizations have come to accept this bureaucratic burden as a necessary evil to ensure task accomplishment. The complexity of both the hardware and software (illustrated earlier) warrants system specialists with narrowly defined job responsibilities. One expert familiar with the costs of the personnel states:

> ...the human costs of running the mainframe are as high as its other costs. And despite the high expenditure on what tries to present itself as a 'luxury' product, one is still faced with inflexibility. The human bureaucracy in the large mainframe installation actually *adds* to the rigidity inherent in the underlying system. [Ref. 4:p. 42]

The isolation of the MIS department behind the glass house contributes to an environment that makes it easy for computer specialists to be insulated from real-world business requirements and unresponsive to end-users. Because of this and already backlogged application development and maintenance, new systems requirements often are not met. With the desktop technological advances, new development tools available, and the higher end-user experience levels, the gap between the mainframe and PC appears to be widening. The market demand is for GUIs, 4GLs, and OOP among other things, where the desktop often exceeds the mainframe in capabilities. Organizations are realizing that the mainframe bureaucracy is costly in more than one way. Not only are the costs of maintaining existing mainframe hardware and software high, but the opportunity costs of not advancing with current software trends are equally significant.

2. Evolutionary Changes Through Desktop Computing

Though the mainframe computer has been the traditional computing architecture of the past, the desktop revolution is charting a new course for the future of computing. The rapid changes and availabilities of opportunities is not only overwhelming, but also confusing. Nonetheless, the new-found power currently available on the desktop has created a technological momentum and an array of architectural options that is mind boggling. Undeniably, the trend has taken over the industry and is changing the way organizations are coping with their methods of processing information. This section will analyze the characteristics of the desktop computer that have spurred this revolutionary change, and will explore the evolutionary IS architectural options and trends that have resulted from it.

a. Defining the Desktop Computer

With the move towards desktop computing, it is important to define some related terminology a little more explicitly for the sake of clarity. When referring to desktop computing, this term refers inclusively to personal computers (PCs) and what are commonly called workstations. Workstations usually have a true multitasking operating system, and are accompanied by high resolution display cards and monitors and a math coprocessor.

Workstations have also been the machine of choice when it comes to scientific applications. In contrast, PCs have commonly been the choice for business purposes. Though these two classes of computers are quickly merging together and the differences inevitably are blurred, Figure 3 illustrates a distinction that Gartner Group makes that may be helpful in clarifying ambiguous definitions and for understanding the differences (subtle and changing as they may be) between the PC and the workstation.

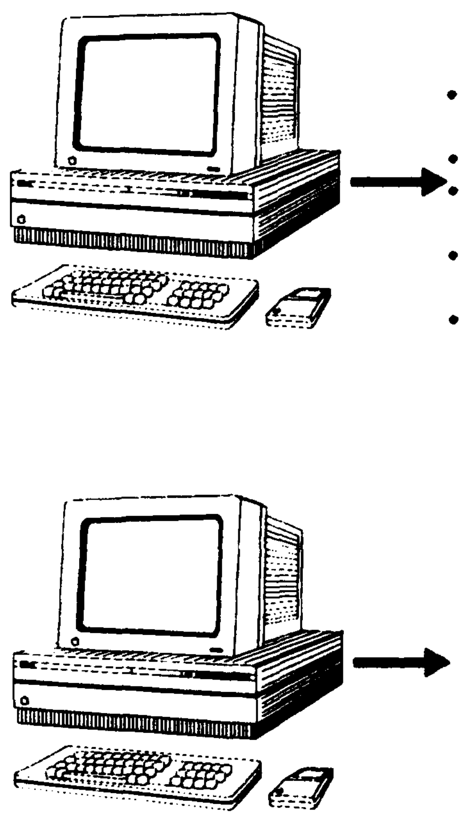


Figure 3: Differences between the PC and Workstation [Ref. 8:p. 1]

b. Perceived Advantages with the Desktop

The pace of change in desktop computers has been staggering: within the past decade microcomputers have advanced their lot from being hobbyists' toys to become the centerpiece of computing. New and more powerful PC hardware and operating systems are promising and delivering performance characteristics, security features, and multiprocessing capabilities that have most often been associated with minicomputers and mainframes. Besides rivaling performance features of more traditional systems, desktop computers are also offering four key attributes often not found with their larger counterparts: cost savings, desktop control, technological flexibility, and responsiveness to business changes [Ref. 9:p. 11].

(1.) Cost Savings. Without a doubt, cost savings is the biggest driving force behind the desktop revolution. The many suppliers and the resulting economies of scale have produced an intensely competitive market with attractive promises of cost savings that have been irresistible to both small and large businesses. One aspect of cost justification has always been to compare and analyze the hardware's price-to-performance ratios. For example, the price per MIPS—as discussed later, not the best basis for comparison—on an IBM 3090-600 mainframe has been quoted at $130,000; at the desktop level, the price per MIPS on the RS/6000, IBM's RISC workstation, is roughly $500 [Ref. 6:p. 4].

Beyond this, the desktop computer market offers organizations cost savings in many other areas. From software development to software maintenance, desktop computing offers not only potentially huge savings, but also a greater variety of options. In terms of scalablity, instead of risking huge corporate investments in untested mainframe projects, organizations may buy only as much computing power as they currently need. Through the use of distributed networks, organizations can modularly add to their infrastructure as requirements dictate.This added benefit manifests itself both when organizations have limited funds to invest in start-up IS projects and when systems grow older and need to be replaced by more capable technology. The architectural options provided by networking and the creation of distributed systems allow for the computing systems to be augmented incrementally, a workstation at a time. Finally, in terms of personnel, great potential for cost savings exists in terms of a more productive work force within a less bureaucratic environment—a function of greater desktop control.

(2.) Desktop Control. One of the readily apparent advantages of desktop computers in a decentralized environment is that this environment brings computing power closer to the end-user. As such, there is no longer as much a dependency on the technicians behind the "glass house" to solve logjams in the system. With this greater degree of empowerment, the more educated end-user can better control the IS infrastructure and help anticipate *and make* changes. This customization of the computing environment helps to foster a greater sense of ownership among end-users; the increased motivation and involvement of users can also increase overall productivity.

Empowerment at the end-user level and subsequent desktop control does not come merely through decentralization. It is also the case that increasingly innovative and sophisticated developments are taking place on the desktop at a much faster pace than at the mainframe level. Industry trends and market dynamics are adapting desktop computing features to meet the needs of their end-users. End-users have demanded a shift away from the character-based interfaces; the graphical user interface (GUI) is becoming the norm. End-users have sought technical flexibility to control their desktop interface, giving them a standard "look and feel" that matches their intuition. The so called WIMP interface—Windows, Icons, the Mouse, and Pull-down menus—is becoming the de facto standard. Voice commands and handwriting interfaces are currently making their debuts. Again, these trends are introduced and becoming standardized at the *desktop level*. Desktop computers, not traditional mainframes with dumb 3270 terminals, represent the leading edge and the future in this area—a trend with significant underlying implications.

(3.) Technological Flexibility. Technological flexibility refers not only to the greater desktop control as discussed above, but more so to the variety of hardware and software options available in a desktop environment. The move away from the traditional mainframe environment, often regarded as too structured, "stove-piped," and proprietary, has created a movement to the other extreme at the desktop level. IS professionals have not wanted to be locked in to a monolithic structured environment, so the move has been towards networks linking a host of heterogeneous desktop hardware and software elements. IS professionals have not wanted to be at the mercy of one vendor, and their demands have met with a market-wide race towards development and implementation of the almighty "open system." End-users have not wanted to rely on technicians behind the glass house, so they have been empowered with applications tools such as 4GLs and other end-user centered applications. The call, in short, has been for technological flexibility at all levels, and the desktop market has been the first to respond to the need.

(4.) Responsiveness to Business Change. This fourth attribute of desktop computing is really an effect of its greater cost savings, desktop control, and technological flexibility. The new challenges of the 1990s have introduced new business imperatives that demand technological flexibility in the IS infrastructure. As organizational changes occur to reflect a changing business environment, so too must the IS infrastructure adapt to efficiently accommodate that change. All too often, "information systems have cast specific ways of doing business in concrete, not allowing a company to change its operating procedures or move into new lines of business as quickly as management would like" [Ref. 1:p. 296]. Thus, traditionally centralized and inflexible IS systems (most often epitomized by the mainframe) may not only be inadequate in this new environment, but could represent a serious liability. Many believe desktop computing, linked to local and wide-area networks to create a heterogenous distributed architecture, provides a more viable alternative to this inflexibility. According to one IT consulting group:

> Resources must be capable of dynamic response to increasingly volatile competitive, market, and economic environments. Moves toward a different paradigm of IS deployment has begun in many organizations... Increasing power and sophisticated of distributed computing systems provide one of the elements of this paradigm. PCs are bringing more powerful, user-friendly IS tools to the individual... small systems and servers, as well as networking developments make it possible to offer new types of distributed solutions. These allow IS to be more closely mapped to business processes at all organizational levels in order to provide more flexible and responsive tools as business needs change. [Ref. 11:p. 1]]

D. THE RESULTING PARADIGM SHIFT: DOWNSIZING

Inspired by the promises of the desktop revolution and the opportunities of new architectures, there is no doubt that IT is in the midst of profound change. A fundamental shift is occurring in the way information is being processed, a move that is most often described as an exodus away from the traditional mainframe towards architectures characterized by networks of smaller, heterogeneous desktop personal computers and workstations. This trend, characterized by the shifting of processing from "big" systems to "smaller" systems, has been labelled "downsizing" of information systems. Truly a paradigm shift, the new opportunities offered by heterogeneous networks of powerful desktop computers forces IS professionals to re-evaluate traditional methodologies of processing information in light of new systems that are governed by different sets of rules.

After many years of practice, IS had started to get accustomed to managing computing on monolithic systems. IT was getting manageable, problems somewhat predictable. That stability is now being overthrown with the introduction of the powerful desktop and distributed, heterogeneous, and more unmanageable systems. The desktop computing environment represents a radically different world from its monolithic predecessor with new software tools and computing techniques to operate in a distributed, multivendor, software-centric environment that focuses on the end-user. This new form of computing will have pervasive effects that may require organizational re-thinking of traditional computing methods, redeployment computing resources, and re-training personnel. IS professionals will be forced to respond to a whole new set of management and technical challenges in this increasingly complex environment. Thus, while downsizing may offer a vast array of new opportunities, this paradigm shift does not come without an equal array of barriers and pitfalls.

Some IS professionals have criticized the labelling of this movement as "downsizing" as too restrictive a term and have suggested adoption of "rightsizing" as a more appropriate term. Rightsizing implies choosing the most appropriate computer—regardless of size—to best suit an organization's processing needs. Another term, "smartsizing," has also been commonly used with the same connotations as rightsizing. Still, others prefer use of the term "upsizing" to describe moving networked personal computers and workstations to an environment with more robust features. This could imply actually moving to larger systems or simply to servers with better capabilities in areas such as security and backup.

Whatever the most appropriate term, there is no doubt that increased processing power at all levels (but perhaps most noticeably at the desktop level) has created the opportunities to think of new ways of processing information. It is necessary, however, to formally define a vague term to clarify its connotations. For the purpose of this thesis, I will adopt the Gartner Group's definition of downsizing:

> Downsizing is the redeployment of information systems from a traditional mainframe to a new architecture, with the primary goal of reducing total IT expenditures across the enterprise. Downsizing generally refers to the migration of processing from traditional mainframes to less expensive alternative mainframes, to midrange systems for conventional application processing—e.g., batch or on-line transaction processing (OLTP)—or to LAN-based networks of midrange and PC-based servers. [Ref. 12:p. 4]

Three essential ingredients contribute and fuel the downsizing trend: recognition of information as a strategic asset, realization that the computing architecture is the essential means of defining an organization's "information strategy," and the belief that the promises of smaller, increasingly powerful computers are the key to fulfilling that information strategy. In short, many CIOs have seen the "writing on the wall" and are embarking on this new voyage. Because technology is changing so rapidly, no one can be certain where the "voyage" will end. What has become too apparent to most organizations, however, is that the future is away from the traditional mainframe—and organizations are getting on the downsizing train as soon as possible before it is too late.

One recent downsizing survey conducted by the Gartner Group at their Midrange Computing Strategies (MCS) Conference polled attendees about their downsizing plans. The respondents represented U.S. corporations with a total data processing budget of close to $5 billion. Indications of the strength of this paradigm shift are evident in the results:

- 74 percent of the respondents indicated that they downsized applications in 1993.
- 96 percent indicated they were planning to downsize in 1994 or 1995
- To contrast this with previous trends, only 60 percent of those surveyed in 1992 indicated that they were planning or implementing a downsizing project. [Ref. 13:p. 6]

III. ESTABLISHING THE APPROPRIATE ARCHITECTURE

The previous chapter discussed the more traditional roles that the mainframe and PC have played in organizations.As the term "rightsizing" implies, however, the downsizing paradigm shift does not necessarily entail establishing an architecture limited to the centralized mainframe architecture or one that is exclusively built around a network of desktop computers. Computing components can be bought, but a computing architecture must be *built* to optimize processing capabilities. Processing power does not necessarily have to be the mere sum of its individual computing components. New architectural tools are helping to synthesize disparate computing components and are enabling more effective and efficient forms of processing. Indeed, the momentum that has been driven by technological advances has unharnessed traditional mindsets and unleased new architectural ideas. This chapter will discuss those new architectural choices and the tools that serve as their building blocks. A final section will discuss migration strategies for migrating from traditionally centralized mainframe architectures to client/server type environments.

A. ARCHITECTURAL DECISIONS

Before the advent of the powerful desktop personal computers in the 1980's, almost all computers performed batch and on-line processing on the traditional mainframe with the aid of "dumb" terminals. The glory days of the mainframe and its future faded dramatically with the introduction of the personal computer. Innovative technology over the years has enabled today's networked desktop computers with processing capabilities that claim to rival those of the older, monolithic host platforms. Besides impressive performance measures, desktop computing now offers the advantage of user-friendly features such as intuitive graphical user interfaces (GUI's) on a platform much closer to the user—for what first appears to be at a vastly more cost-efficient price.

Developing architectures, however, do not necessarily force IS departments into an either/or decision between mainframes and networked desktop systems. New architectures, such as client/server, are helping break the established boundaries of IS and are allowing platforms to take advantage of mutual processing capabilities and exploit each other's strengths. More flexible architectures, that include integration of current systems into new architectures, are helping organizations respond to an increasingly global and competitive environment in new and more effective ways.

1. The Client/Server Model

The client/server model, perhaps the most important architectural by-product of the downsizing trend, was developed as a direct result of the increasing power and decreasing costs of desktop systems. Simply put, the objective of client/server is to maximize efficiency and effectiveness of computing resources by allowing various elements to specialize at what they do best.

One of the most pervasive new trends currently dominating the IS world, client-server is a term that is quickly becoming an industry buzzword to describe the new movement in distributed computing. Like so many buzzwords, however, the term client/server has been used so often and in so many different contexts that it has come to mean different things to different people. With so many degrees of distributed computing and varieties of implementation, the terminology associated with client/server has become somewhat confusing.

It appears that there are two schools of thought in this area. One school uses the terms *distributed computing*, *cooperative processing*, and *client/server* loosely and interchangeably to describe the client/server architecture, while the other tries to define technical and semantic differences between the three. In general, however, most IT professionals would probably agree that client/server computing is a form of distributed computing and involves a degree of cooperative processing. Indeed, one IS professional tries to clarify the terminology with the following assertion:

> A distributed system is defined as a computing environment in which processes and/or data are distributed throughout an organization with the requirement that a degree of interaction between systems is required to complete processing the workload. *It is to this category that the term client/server relates. There is no particular significance (other than semantic) in distinguishing client/server from distributed systems.* [Ref. 14:p. 32]

The client/server term essentially tries to add a little more precision to the above definition of distributed systems by defining the specific roles of computers within a distributed environment. Again, however, IT professionals have trouble standardizing this term; one Chief Executive Officer of a mainframe software company believes "client/server is a term that has been overused and is confusing customers because different people are using it in different ways" [Ref. 15]. Most IS professionals agree that the architecture splits up the processing between a client, usually a desktop, and a back-end server. The degree to which the processing is split up is where the definition becomes vague. Typical definitions of client/server resemble that given by a well-known IS consultant:

> Client/server architecture splits an application into a front-end client component and a back-end server component. On the front end, the **client** part of the application, running on a workstation, gathers data from the user, prepares it for the server, and issues a request to the server. On the back end, the **server** waits for requests from its clients. When it receives a request, the server processes it and returns the requested information to the client. The client then presents the data to the user through its own interface. [Ref. 16]

This definition is somewhat incomplete. A more comprehensive definition would include a discussion of the client and server roles in data presentation, application location, and data management. The Gartner Group consulting service, recognizing the varying degrees of client/server uses Figure 4 to illustrate the different models and "shades" of client/server computing:

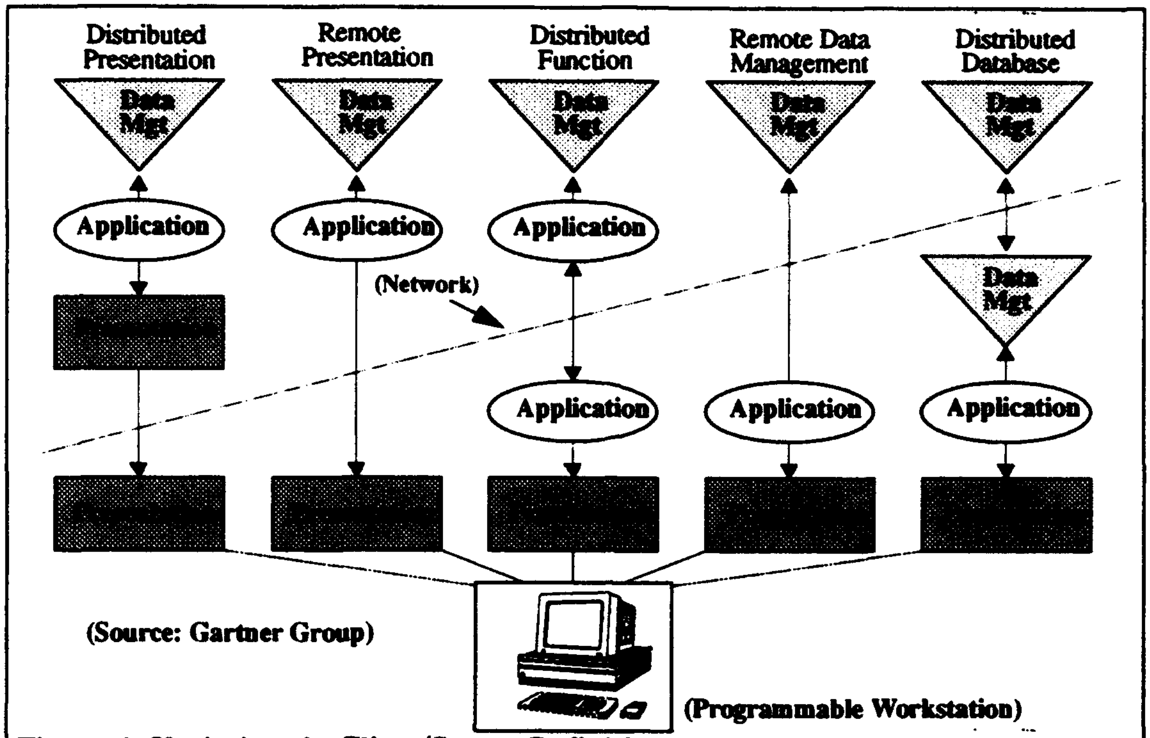


Figure 4: Variations in Client/Server Definitions [Ref. 9]

- Distributed Presentation: Also known as "screen scraping," the programmable workstation (client) is run by a host (server) based application that alters the dull 3270 character-based format to a more friendly GUI interface. Additionally, the workstation is prompted by the host to collect user inputs (using the GUI format) and interact with the host. Some IS professionals do not regard this or the remote presentation (see next paragraph) as "true" client/server.
- Remote Presentation: Again, the desktop workstation acts like a client in this situation. In this case, the desktop runs a presentation application capable of receiving messages from the host-based server and is able to utilize more involved GUI features such as menus, windows, and dialog boxes.
- Distributed Function: Commonly referred to as the "peer-to-peer" model, the application is split between the between the desktop and the host (not necessarily a mainframe), with each acting as the client (requester) or server (provider) at various times. Processing may occur concurrently on both platforms, with control of the application also shifting between the different nodes.
- Remote Data Management: In this case, the application wholly resides on and is driven by the desktop computer acting as the client. The client submits data queries or updates to the server, which responds to the request.

- Distributed Database: The presentation functions, application, and some data management reside on the desktop computer (client). Queries are submitted, as necessary, to the server. Here, data may be distributed on multiple nodes throughout the infrastructure. [Ref. 9]

Thus, it becomes more clear that the variance in IS professionals' definitions of the client/server model occurs *as a result of the varying degrees of recognition* of client and server responsibilities. One *Computerworld* survey reveals this disparity of IS professionals' definitions of client/server and illustrates how their definitions differ directly with the perceived processing roles of the client and the server. The survey was based on the responses of 219 informations systems professionals who run applications in the client/server environment. The following results were obtained:

Table 3: "HOW ARE YOU DEFINING CLIENT/SERVER?" [Ref. 17:p. 8]

19%	24%	33%	22%	2%
Client Presentation	Client Presentation and some business logic	Client Presentation and some business logic	Client Presentation and some business logic	Client Don't know
Server Everything else	Server Additional logic, data management and some distributed systems	Server Additional logic, data management and fully distributed systems	Server Data management	Server Don't know

With four almost equally divided definitions, the survey effectively illustrates the disparate perspectives of the client/server paradigm and helps explain some of the customer confusion. At a minimum, however, the survey in Table 3 indicates that all IS professionals at least agree that the GUI interfaces of the intelligent desktop make it the platform most capable of handling the presentation functions. At the other extreme, many organizations prefer that the client handle not only the presentation-related processing, but most—if not all—of the application logic. [Ref. 17:p. 8]

Despite the varying opinions of where the processing takes place, what is clear is that many organizations readily recognize the potential benefits of the architecture. As DoD and corporations reengineer and redesign their organizational structure and information processes, IS structures are reevaluated and realigned in accordance with strategic and tactical objectives. Client/server is often argued to be the most logical means of realigning the IS architecture because it exploits the same perceived advantages of desktop computing (such as the reduced hardware and software costs with increasingly powerful performance capabilities—as described in the previous chapter) while creating an environment that is responsive to business needs. Other unique and commonly cited client/server's advantages include:

- System flexibility: The increasing standardization of protocols and "open systems" permits ad-hoc integration of disparate platforms. As requirements change, the client/server architecture provides for the easy integration of additional nodes. Furthermore, IS departments can integrate old technology with new technology—utilizing sunk investments instead of jettisoning old equipment. In this sense, the client/server methodology can be implemented in a non-disruptive, self-paced, manner.
- User-centricity: Applications development tools (e.g. 4GL's) and data manipulation methods focus on the user. For example, a user's database query is responded to, theoretically, with "seamlessness." Physical location of application or data is not pertinent.
- Vendor independence: Client/server unleashes IS departments from the dependence on vendors' proprietary hardware and software. As more protocols and systems truly become standardized and open, users can select the "best of breed"—the system that provides the desired functionality for the least price.
- Reliability: Though one machine may go down, in a client/server environment there is enough redundancy and machine independence to allow the business to continue normal operations.

Despite the disproportionate focus on the personal desktop computer in client/server, the mainframe and minicomputer can also play important and needed roles in this architecture. Many organizations have been either strictly PC-centric or mainframe-centric; client/server offers a blend. As suggested earlier, client/server is a flexible architecture that allows for the integration of current systems in a distributed environment. Mainframe and minicomputers still have many desirable characteristics that workstations will have difficulty matching.

2. The New Role of the Mainframe

Many IS professionals believe that the move toward client/server need not exclude the mainframe. Distributed computing can mean that new technologies can supplement rather than displace older ones. With the client/server model, a network composed of main ames (to include mid-range computers) and powerful desktop computers can all play critical roles. Performance can be optimized by the shifting and redeployment of data resources and computing power according to task requirements and system strengths. Many IS consulting groups believe that the mainframe's role will evolve this decade in new ways that are synergistic with the boom in distributed computing. Taking advantage of the mainframe strengths and the current trends, they generally agree that the future of mainframe computing will be as an enterprise hub with central roles in the areas of data management and networking management.

a. Data Management Hub

Data management has always been a core strength of the mainframe. With the mainframe hardware, operating system, and software optimized for management and movement of large volumes of data, the mainframe has been commonly selected as the corporate hub for mission-critical applications. Because of these sophisticated data handling capabilities and the lack of feasible alternatives, the mainframe has historically dominated—and consequently saturated—the data management market with its systems and applications.

Some estimate that corporations around the world have spent over $750 billion on *software* applications for IBM and IBM-compatible mainframes alone. Furthermore, the performance capabilities of the mainframe are increasing at an impressive rate along with the desktop. According to one report, the aggregate number of MIPS on mainframes almost tripled between 1988 and 1991 to from about 125,000 to about 364,000. [Ref. 18:p. 29] Some believe, however, that this market base and performance lead will slowly be eroded by product developments that offer high-performance, low cost alternatives that can potentially rival the mainframe. MIPS capacity on the desktop, for example, grew much more than three times during that same period, thus reducing the mainframe's *relative* capacity. This trend is likely to continue—some absolute growth in mainframe capacity, but an erosion in its relative lead. Unless alternative platforms can prove to be significantly more cost-effective though, the mainframe's role in data management is unlikely to change dramatically in the near future. According to the Gartner Group, the "MVS/390 is the logical candidate for the central data server and as a platform for data warehousing. After all, it is home to the bulk of enterprise production data today, and therefore it involves no effort to leave the data there" [Ref. 7:p. 26].

b. Network Management Hub

The distributed computing environment, current lack of uniform network standards, and increasing transaction volume has increased the burden and complexity associated with managing multiple and diverse local area networks (LANs). Managing these LANS separately is difficult and costly. Given its track record and promise of more powerful mainframes, IS professionals believe that the mainframe's current network management role can be extended to service the other types of networks (see Figure 5).

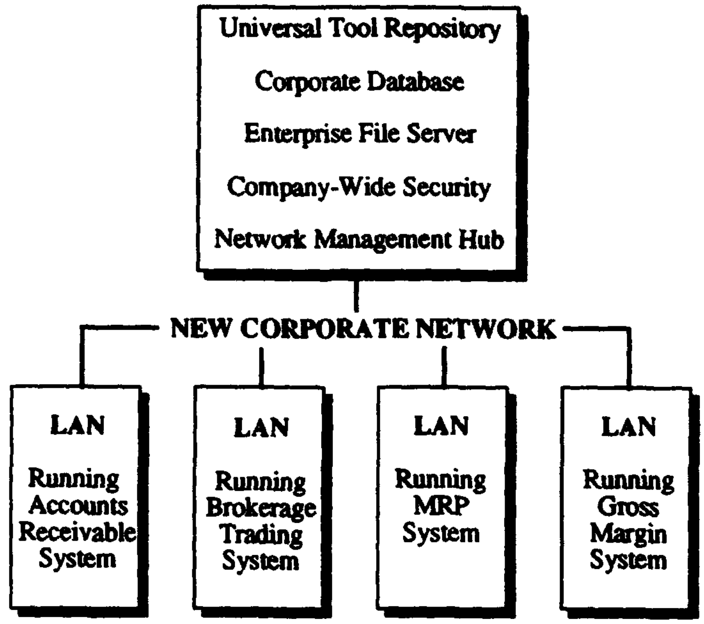


Figure 5: Mainframe as a Network Management Hub [Ref. 6:p. 47]

Currently, IBM's MVS widely serves as a management hub for its existing proprietary Systems Network Architecture (SNA). In addition, SNA backbones are used to carry other LAN traffic throughout the enterprise. Expanding its role as a management hub to other standards-based systems that are growing in this heterogeneous environment (e.g., TCP/IP, NetWare and IPX/SPX) is a logical extension to its current role. Managing the networks from one central mainframe would reduce complexity and redundancy by consolidating monitoring and management of the systems. Because it is already often used as a focal point for communications switching and is built to provide high bandwidth, the mainframe's role and capabilities make it a suitable and appealing choice in the client/server environment.

Other essential features that a network management hub must guarantee are data security and back-up. Ensuring that mission-critical data are secure and backed-up becomes significantly more difficult as the numbers and diversity of LANs increase. Using its sophisticated security mechanisms and storage management, the mainframe can act as a central management point to ensure that only authorized users can access dispersed databases and that those databases themselves are intact.

B. ARCHITECTURAL BUILDING BLOCKS

As important as the new architectures themselves are the tools needed to design, develop, implement, and maintain those architectures. The technological advances in desktop hardware have not come without equivalent advances in related software. From increasingly more powerful operating systems to new software development approaches, the desktop computer continues to rapidly progress with impressive innovations. Software developers, driven by sheer number of PCs and the resulting huge demand for related software products, are responding with quick delivery of new and inexpensive software products to meet changing business needs and architectural requirements. The high volume/ low price market of desktop software is accelerating greater progress and innovation of software products than is possible in the low volume/high price mainframe software market. The low prices of software packages and innovation of software tools can, by themselves, make downsizing to smaller systems a very attractive option. This section will briefly discuss those software building blocks that help build architectures like client/server—and provide additional momentum for the downsizing trend.

1. Increasingly Sophisticated Operating Systems

As the desktop computing has evolved to play a more central role in corporate computing, their operating systems have simultaneously advanced to run more complex applications and assume new duties that had been typically relegated to larger systems. Not satisfied with the PC only playing the word processor and spreadsheet roles, vendors have upgraded the technological capabilities to provide a robustness that had long been integrated into the minicomputer and mainframe.

Currently, the more powerful 32-bit operating systems that are most competitive within the client/server market include UNIX, OS/2, and Windows NT. Offering true multitasking, multithreading, and multiprocessing, these systems can tap the power of 386, 486, Pentium, and PowerPC microprocessors and support applications with 32-bit address space capabilities. According to Gartner Group, "current PC operating systems, such as Windows NT, bear more resemblance to VMS (or even MVS) than they do their ancestors (e.g., DOS) by implementing virtual memory, multiprogramming, pre-emptive multitasking, 32-bit addressing and symmetric multiprocessing" [Ref. 7: p. 7]. Designed to support the burgeoning distributed computing environment, these operating systems not only enable cooperative processing over multiple systems, but also can provide a degree of data and networking integrity and security. Impressive systems management functions and productivity tools for applications developers make the operating systems extremely attractive in the client/server architecture.

With the wide variety of competing operating systems, a major question for downsizing corporations is which one will be the desktop OS of the future? Currently, Microsoft Windows and Unix appear to be dominating the market, with Windows NT positioning itself to gain a large part of the market share. Windows NT is trying to accomplish what Unix attempted a decade ago by providing a portable operating system across different hardware architectures. Microsoft, however, is going a step farther by attempting to provide a comprehensive set of standards with its WIN32 based operating system, GUI application programming interfaces (APIs), applications, DBMSs, languages and Windows Open Services Architecture (WOSA) interoperability features [Ref. 7:p. 45]. Although Windows NT and Unix are certain to survive any operating system "war" and are likely to be formidable players in the future, it is also likely that other alternatives will remain strong market competitors. As such, trying to predict an "OS of the future" to use as a basis for establishing a computer architecture is neither an appropriate or fruitful exercise.

2. Graphical User Interfaces (GUIs)

Another manifestation of the increasing sophistication of the operating systems are the impressive graphical user interfaces (GUIs) that they support. The GUI interfaces available for the different hardware platforms and operating systems are becoming more standardized; the major GUI interfaces outlined in Figure 6.

Microsoft	Windows/Windows NT
IBM (OS/2)	Presentation Manager
Sun (Unix)	OpenLook
OSF (Unix)	Motif
Apple	Macintosh

Figure 6: Major GUI Interfaces

Providing a consistent seamless, interface for client/server related applications and services, GUIs are replacing traditional character-based user interfaces that are acting as a client front-ends. The consistent screens and menus help create a familiar and easy-to-use "look and feel" that helps to mask the complexity and heterogeneity of the client/server architecture from the end-user. GUIs help empower the end-user, allowing more intuitive screens that allow easy retrieval and manipulation of corporate data—reducing training time and expense while also boosting productivity.

3. Database Management Systems (DBMSs)

Once the responsibility of application programs, the role of managing and integrating mission-critical data spread throughout a distributed environment is now the job of increasingly sophisticated database management systems. Based on the relational data model and the Structured Query Language (SQL), current DBMSs are able to select and combine diverse data elements located in diverse and different systems. In typical configurations, the data management logic resides in the server and the applications while control logic remains in the client. Sharing processes and exchanging information seamlessly from an easy-to-use client interface is critical to the DBMS. Advances in DBMS technology allow some systems to retrieve and change data in multiple database servers.

Two problems that have hindered more rapid progress of DBMS' and total integration of all databases are (1) different SQL standards resulting in incompatibility problems and (2) legacy data that are stored in non-relational files. Tools have been developed, however, to access and translate data resident in these platforms:

> ...front-end tools such as Information Builders' Enterprise Data Access (EDA)/SQL are making possible end-user access to legacy data. These SQL front-end software products and real-time language interpretations enable restructuring in relational terms of legacy data for knowledge workers and make real the promise of universal data access... EDA/SQL provides data access to more than 50 different data sources and file structures on over 35 disparate hardware platforms. [Ref. 19:p. 11]

4. Middleware

Heterogeneous, distributed architectures such as client/server require interoperability across dissimilar operating systems, communication protocols, and databases. *Middleware* is the enabling layer of software that accomplishes this. Residing between the business application itself and the network infrastructure, the goal of middleware is to unite independent local operating systems and transform them into a structure that creates the illusion of being part of a single, monolithic system. Also nicknamed "glueware," this software establishes virtual connections across hardware boundaries, "gluing" together multiple users, workstations, application programs, CPUs, operating systems, network protocols, file systems, and other resources. A more precise Gartner Group definition is:

> The layer of software between the end-user application code and the operating system. Examples include DBMSs, UIEs (User Interface Environments) and TP (Transaction Processing) monitors. The end-user application invokes middleware services via APIs (Application Programming Interfaces), and sometimes invokes operating system services via different APIs. A middleware component (which is merely an operating system application) also uses an API to call for operating system services.

In a typical IS application of the 1990s, the less frequent of the three cases is the one where application programs directly invoke operating systems without passing through a middleware component. [Ref. 20:p. 10]

A difficult concept to define precisely, it is helpful to examine what middleware is ultimately trying to achieve. Simply put, middleware allows one procedure to locate another procedure anywhere on the network and exchange messages and information. This exchange is permitted to occur between sender and receiver regardless of physically separate locations, underlying communication protocols, and different operating systems. In accomplishing this objective, middleware is also expected to provide the ancillary services of data management, presentation, communication, and management services depicted in Figure 7.

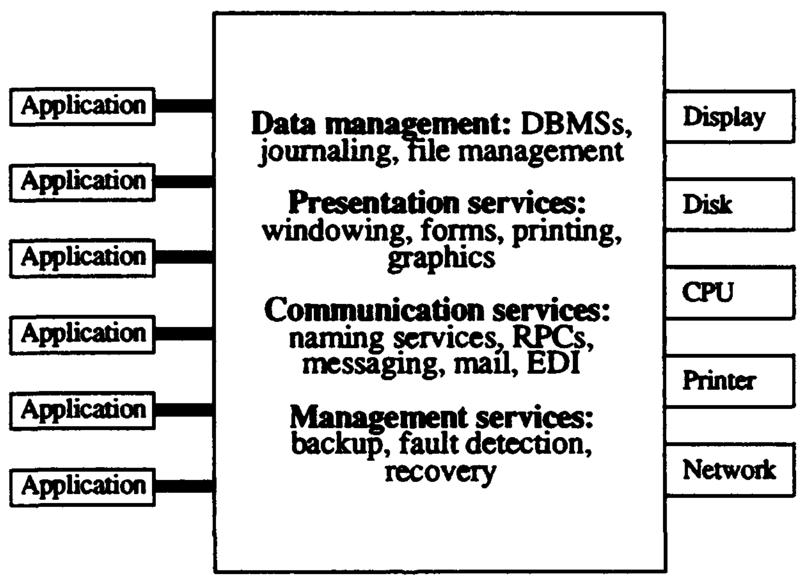


Figure 7: Middleware for Network Computing [Ref. 7:p. 9]

Organizations such as the Open Software Foundation (OSF) have helped to standardize some of the middleware services through establishment of models such as the Distributed Computing Environment (DCE). The middleware goal, to obtain a virtually homogeneous architecture through a system of transparent software services, still remains, however, to be the greatest challenge of client/server and distributed computing.

5. Application Development Tools

Many transaction processing and management information needs are not being met in the current mainframe environment for the simple reason that the cost of providing software to meet new requirements is too high. The high expenses result not only from software development, but also from the need to maintain applications as market and business requirements shift. The consequential application development backlog has encouraged developers to seek relief through other avenues. The downsized environment, many believe, offers viable alternatives through both the cost-effective purchase of increasingly sophisticated off-the-shelf packages and new software development methodologies and tools. This subchapter will highlight the impact of some of the new methodologies and tools to include the notion of rapid application development (RAD), 4GLs, and CASE tools.

a. Rapid Application Development

Rapid application development (RAD) is a term that encapsulates the trend in software development that digresses from the classic style of software development—best epitomized by the "waterfall" model—towards a methodology that dramatically accelerates delivery of software. The problem with the classic methodologies like the waterfall is that the process requires comprehensive designs, formal reviews, and cumbersome documentation that often tie down progress in bureaucratic inefficiencies. RAD techniques discourage this bureaucracy by eliminating formality and exhaustive reviews and encouraging constant developer and end-user feedback as the system is being written. RAD methodologies include prototyping and incremental development.

Prototyping, according to some experts, is used for the purposes of user interface design, performance modelling, and assessment of functionality [Ref. 6:p. 229]. When developing a product, it is difficult for programmers to accurately assess end-user needs and requirements and determine realistic system performance measures. By modelling initial designs to the end-user, developers can rapidly determine errors due to misinterpretation and miscommunication and can also establish initial system performance benchmarks. Because of highly productive prototyping tools, new versions that more accurately meet end-user needs can be built. According to one IT consultant, this process "accelerates failure"—developers can build application prototypes far faster and at far lower cost than they could on the mainframe, giving them the luxury of building multiple prototypes until they find one that works best [Ref. 21].

Incremental development is different than prototyping in several ways. In this methodology, developers build and deliver only part of a system at a time. As end-users identify a requirement, developers *design, program, test* and *implement* it as rapidly as practical. In this fashion, end-users have the ability to gain the functionality of system very quickly. This methodology can only work in systems that do not need to be implemented as an integrated product.

b. Fourth Generation Languages (4GLs)

One tool that plays an integral part of the downsized environment, alleviating the application backlog of the end-user and enabling RAD technology, is the 4GL. 4GLs empower developers by providing an engine that automatically generates program code for both application screens and program logic. Because of these sophisticated capabilities that are sometimes easy enough for the end-user to understand and utilize, 4GLs have even been touted by some as the panacea for the slow pace and high expense of application development.

According to Gartner Group, 4GLs can be classified into three types: "frontware," database-linked, and database independent. *Frontware 4GLs* empower users by adding the capability of rapidly generating front-end GUIs on the client-end to access legacy data. Frontware 4GLs depend on middleware to access the data and handle resolution of all software linkages. In contrast, a *database-independent 4GL* also requires middleware, but not to the full extent that a frontware 4GL does. A database-independent needs middleware primarily to help establish communication links and access to the database. A *database-linked 4GL* is different than a database independent 4GL in that the application development tool dynamically creates links to existing databases. In other words, the middleware component is built into the 4GL application development environment. In the client/server environment where the mainframe maintains a role as a data server, all of these 4GLs can play an important function in extending the life of the mainframe. As new processing requirements and information needs shift, 4GLs can play an important role in responding to changing business requirements. [Ref. 22:p. 2]

Some critics downplay the significance of 4GL technology. Although they acknowledge some 4GLs offer significant benefits, they also believe 4GLs are generally proprietary products that inhibit program portability. They state that 4GLs may be suitable for small system requirements, but that they are generally unsuitable for complex programs requiring intricate data handling *and* that 3GLs are often a much better solution. Although these arguments may be valid in some instances, some very capable 4GLs have already proven their worth in very demanding and complex environments. Ultimately, the suitability of 4GLs very much depends on the particular system architecture and application requirements. The areas where 4GLs are generally recognized to provide increased productivity over 3GLs, however, are shown in the Figure 8.

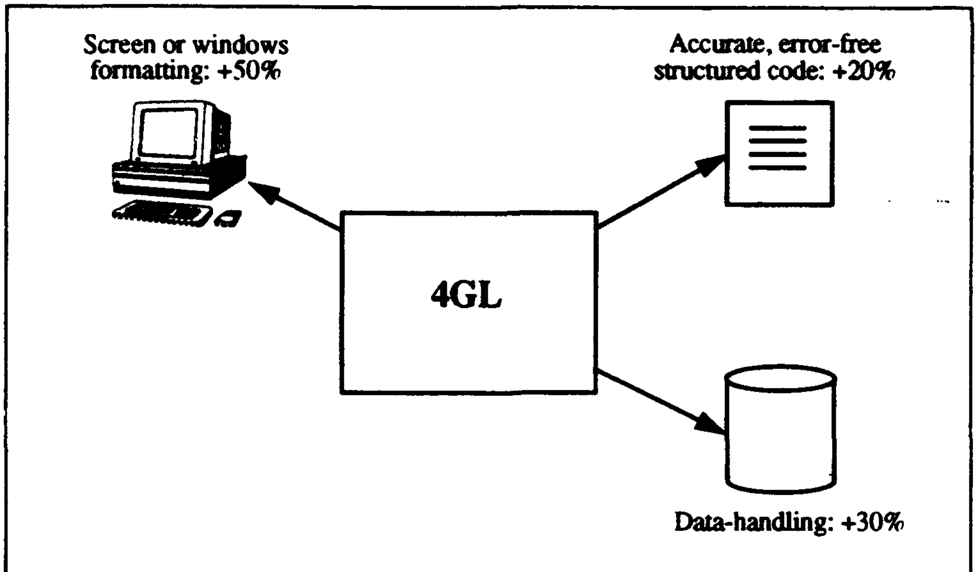


Figure 8: Productivity Increases with a 4GL [Ref. 4:p. 212]

c. *Computer-Aided Software Engineering (CASE) Tools*

Another architectural tool that has been designed to automate large portions of the application development process is a set of CASE tools. CASE tools can play an important role in the application development environment (ADE) by providing a central repository for project planning, management and support activities, documentation, configuration control, and a software re-use *while* facilitating better communications among project members. Such features can be of great value in a complex distributed environment, potentially dramatically increasing productivity and reducing costs.

In the past, however, CASE tools have met with mixed reviews and have not always lived up to expectations. Criticisms have generally been related to the high expense associated with the purchase of CASE tools and the lack of technological maturity. According to one authority, "To be effective, CASE tools must work closely with other development tools such as repositories, code generators, and front-end development tools. Unfortunately, few CASE tools have been able to do this well enough yet" [Ref. 6:p. 223]. Ideally, developers need I-CASE (Integrated CASE) tools that combine upper-CASE tools and lower-CASE tools so that they can be used together to develop network applications. Upper-CASE tools are responsible for the mapping of business processes and data flow designs, whereas lower-CASE tools use those designs to help generate code. Though acknowledging weaknesses in past CASE technology, the Gartner Group expects a "rebirth" in CASE with a new generation of tools and predicts "[a]s CASE vendors take on the emerging new technologies, CASE-driven client/server application development will pick up" [Ref. 23:p. 3].

C. MIGRATION STRATEGIES

Defining a corporate IS objective—such as off-loading the mainframe applications into an open, client/server environment—may be much more easily said than done. Admittedly, client/server and other distributed architectures that offer potentially large cost savings may be a desirable and achievable objective, especially given the advancement in architectural tools. Yet, how is this downsizing actually accomplished? What kind of applications are best suited for an initial downsizing? Moreover, assuming identification of the best application to downsize, what is the *best* method for migrating that code? This section will take a look at the issues behind the ultimate objective, examining options and opinions when selecting the best application to downsize and the available migration strategies.

1. Selecting the Best Application(s) to Downsize

Most IS experts agree that most successful downsizing projects occur by migrating only one application at a time (or portions thereof) and only after selection of suitable "candidates." One general statement often made is that typical legacy systems with high maintenance costs and low mission-criticality are the ideal ones to immediately downsize and should be used as pilot projects. The Integrated Automation (IA) Corporation makes such an argument and provides general guidelines for downsizing applications in Figure 9.

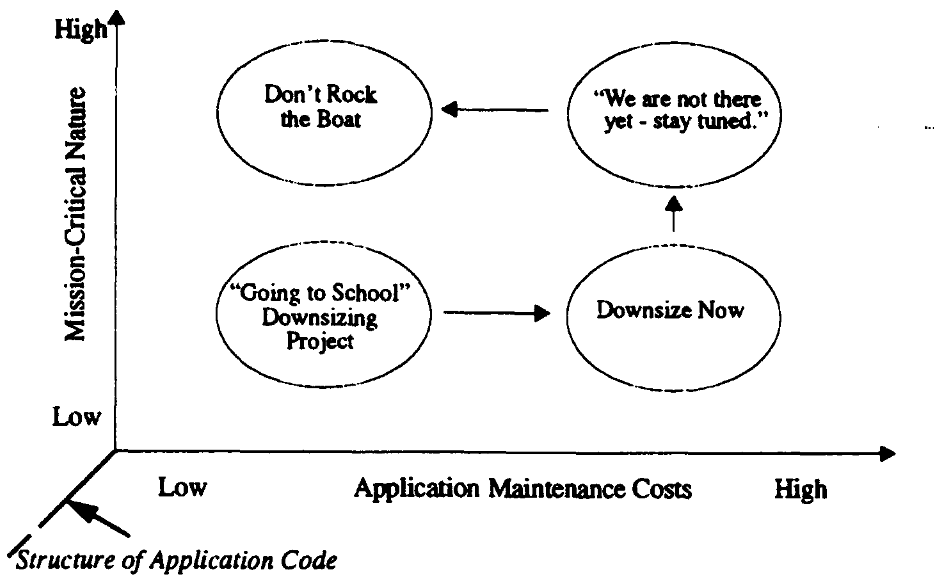


Figure 9: Choosing the Best Downsizing Candidate [Ref. 9]

Certainly, strong consideration is due to issues of mission-criticality and maintenance loads. Governing decision making processes by those criteria alone, however, would be to make decisions according to relatively superficial heuristics, as depicted in Figure 9. Another dimension that is critical to examine when analyzing the feasibility of downsizing an application is the actual structure of the code. It is important to have code that has clearly defined functions and interfaces. Codes with such characteristics will make any re-engineering and re-writing of complex code much less costly in terms of time and money. The structure of the code may also help determine performance parameters in the new environment and the difficulty of separating portions of applications to enhance end-user features. Figure 9 depicts the structure of the application code as a third dimension to consider when choosing downsizing candidates. Common features of code that may lend an application as an appropriate candidate for downsizing may include:

- Interactive processes between end-user and application
- Isolated functional components where application logic can be easily separated from data and user I/O processes.

- Well-defined function interfaces (i.e., one entry and one exit).
- Data structures that are easily separable from application logic (i.e., SQL calls are separated into a logical set of functions).
- New applications—so they can be optimized for the client/server environment. [Ref. 24:p. 3-2]

On the other end of the spectrum, applications that should not be downsized are those where the application and code have characteristics exactly opposite to the ideal candidates. These may include applications where there are few interactive processes, have poorly defined function interfaces, and where data structures are integrated into the application logic. More intuitively, applications that are seldomly used, frequently changed, or about to be replaced are also not appealing candidates for downsizing [Ref. 24:p. 3-4].

2. Migration Techniques

Selecting appropriate applications to downsize is only the first step in implementing a migration strategy. Having determined "what" to downsize, the next question becomes "how?" When deciding on the particular method, Gartner Group recommends first adopting a "global" strategy with regard to the ultimate fate of the mainframe. If the goal is to junk the mainframe in the near future, for example, it can affect the migration technique. Gartner Group believes that organizations need to commit to either (1) growing with their traditional mainframes, (2) fading out the mainframe while preparing replacement systems, or (3) killing the mainframe as quickly as possible. [Ref. 7:p. 32]

With an overall strategy governing the fate of the mainframe, the technique for migrating an application becomes more obvious. Such options generally include:

- *Maintain the status quo*. When the application meets the criteria for candidates that are not good choices for downsizing, leaving the application on the mainframe may be the best option. As modifications and patches are required to meet changing business needs, maintenance continues to be performed in the traditional fashion.
- *Surround and integrate*. With this methodology, the application remains mostly intact on the host. The idea is to build around mainframe and integrate it within client/server architecture. Accessory applications may be added to increase the overall level of functionality. One typical strategy is to simply add a GUI interface

to the client while leaving the bulk of the application processing and data on the mainframe.

- *Transfer*. This requires moving the application to the new platform as inexpensively as possible. The assumption here is that the code is relatively portable.
- *Convert or emulate*. Because the code may not be portable, conversion tools can be used to modify the code to allow it to run on the downsized platform. Another possibility is to emulate the application with a product that emulates current environment. This is a low-cost alternative that retains familiar features and functionality.
- *Reengineer*. In this instance, the old code is still useful and applicable enough to the downsized environment and business processes—the code simply needs to be reengineered. Reengineering code (not to be confused with reengineering business processes) is a software methodology that capitalizes on much of the logic of the old code to help develop more structured, streamlined, and efficient new code. Often the new application is written using the old application as a skeleton and upgrades are added as appropriate. CASE tools and 4GLs are useful tools. This process may be both time consuming and expensive. There is the added danger of replicating inefficient and outdated code, not to mention outdated business functions.
- *Replace*. Business processes may have been reengineered (in this instance, reengineering refers to streamlining of business processes, not code—more about business reengineering is discussed in the next chapter) making the old programs obsolete *or* reengineering the code may be prohibitively expensive in light of other options. The old code is simply discarded in favor of more appealing software alternatives. New tools enable faster, more efficient development of applications. Off-the-shelf software also looks attractive.

IV. ASSESSING THE RISK

Some view the trend that has resulted in the downsizing of information systems as a dangerous fad. Every CIO must have an instinctive fear of their CEO glancing at an IS journal, reading an eye-catching article about a downsizing success story, and receiving a follow-up phone call with the major question of "Why aren't we downsizing?"

The crucial question of "to downsize or not to downsize" has no easy answer. Many questions on a wide range of topics need to be studied and analyzed carefully. Downsizing a major information system can be a risky proposition with significant consequences for an organization. Being able to accurately assess the risks associated with downsizing and to make objective and intelligent decisions to the downsizing question is a critical skill.

For many organizations, business imperatives may pressure downsizing initiatives. Perhaps a move to a new building with new space limitations, cost-saving mandates from above, or new software requirements prompt the initial move—and limit possible options. In such cases, downsizing may not be so much an option as a *directive*. In other instances, an organizations may view downsizing as an *opportunity* to stay ahead of technology and gain a competitive advantage. This organization may have fewer constraints on their options. Whether an organization is downsizing an information system in response to a directive or an opportunity, it is important to properly consider the risks and perform appropriate studies associated with that decision. These studies have been typically labelled *feasibility studies* by systems analysts and form the centerpiece of the systems analysis and the systems development life cycle (SDLC).

This chapter will examine three feasibility studies that are helpful in analyzing the risks associated with downsizing information systems and should be completed during various stages of the SDLC and prior to committing additional resources. These feasibility studies (or analyses) typically include operational, technical, and economic factors. To broaden the scope of the operational analysis to include more organizational implications, the operational feasibility study is re-labelled as an organizational analysis. Additionally, as is often done, the economic analysis portion of this chapter is discussed in terms of a cost/benefit analysis. Because these feasibility studies are completed as an integral part of the systems analysis portion of the SDLC, it is important to first provide a brief overview of the systems analysis process prior to addressing the feasibility studies.

A. OVERVIEW OF SYSTEMS ANALYSIS

Systems analysis is the "study of a current business system and its problems, the definition of business needs and requirements, and the evaluation of alternative solutions" [Ref. 25:p. 9]. Assessing the risk of downsizing is an essential part of the *systems analysis* portion of the systems development life cycle (SDLC). The correct decision to downsize an information system depends on an understanding of the business system and its problems, the end-user requirements, and an understanding of the architectural choices. Systems analysis is a crucial phase of the SDLC. Though many different models exist for the SDLC, all models generally incorporate a phased approach wherein the systems analysis portion plays a key role in initially surveying the project scope and feasibility and ultimately selecting the best solution from candidate alternatives. Figure 10 depicts the phases of systems analysis in one model of the SDLC.

1. The Systems Analysis Process

Systems analysis is a "process performed by many but practiced by few" [Ref. 25:p. 134]. Though it is the crucial process upon which all the subsequent phases of the SDLC depend, it is one that is also too often guided by ill-defined standards and a

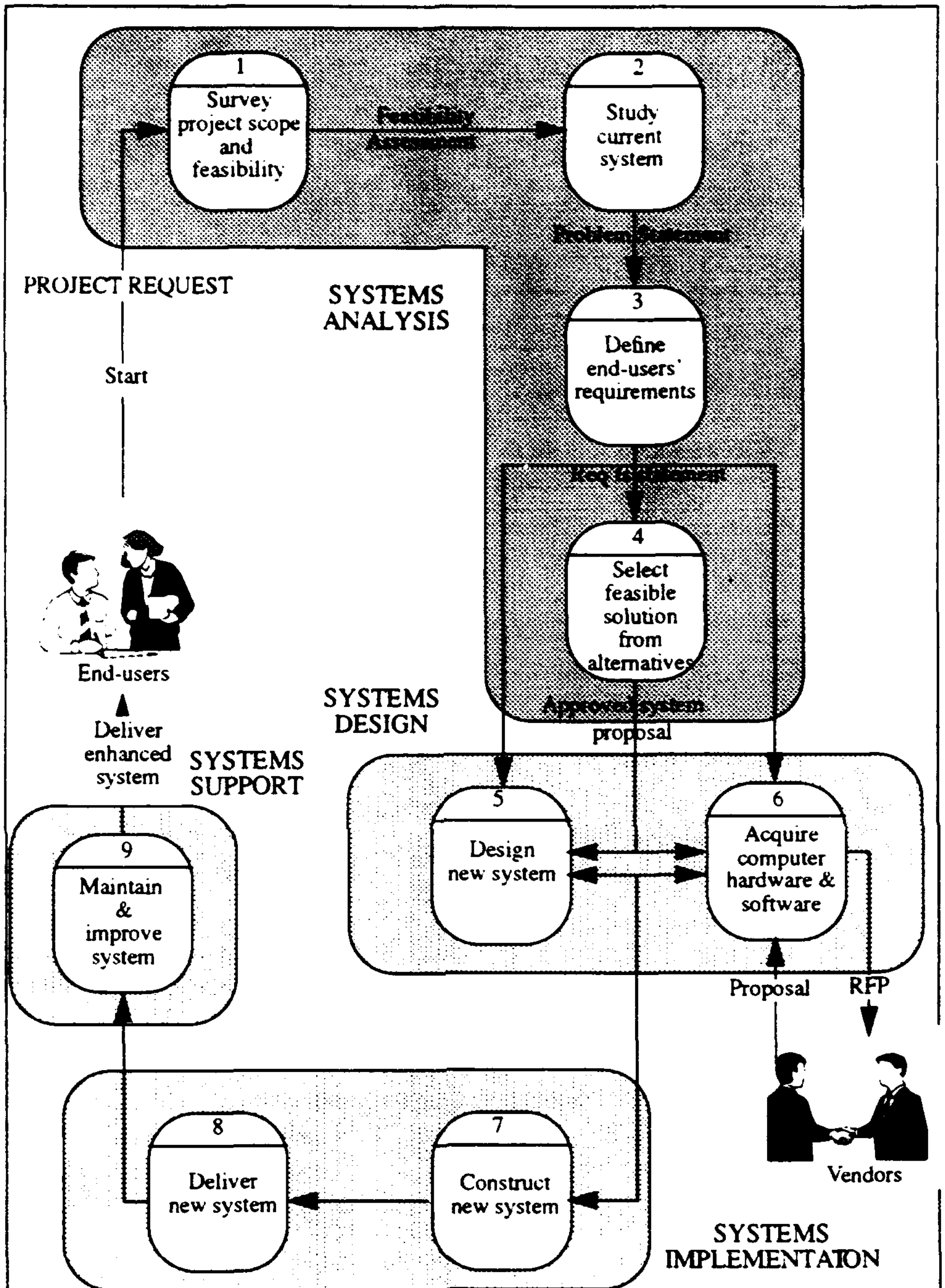


Figure 10: The Systems Development Life Cycle (SDLC)
[Ref. 25:p. 140]

haphazard approach. Good systems design and implementation are dependent on proper systems analysis. The SDLC diagram suggests a more structured methodology to approaching systems analysis, an approach that will enable a more structured analysis and an opportunity to evaluate risks. The four phases illustrated in the system analysis portion of the diagram are:

- Survey. A survey of the project scope and feasibility that serves as a preliminary investigation of the proposed project. Initial parameters of the project are established to include project scope, perceived problems, constraints, possible solutions, and intended goals.
- Study. An establishment of a baseline of the current business needs and system to provide insight to current problems, business needs, and implications of possible solutions.
- Definition. A determination of the end-users' requirements. The emphasis is on *what* the system is supposed to do, not how.
- Selection. The process of choosing the most feasible solution from the available alternatives with a closer look at operational, technical, and economic feasibility. [Ref. 25:p. 167-169]

2. Determining Feasibility

One purpose of systems analysis is to estimate the risk associated with a project. Feasibility studies should be conducted throughout all phases of the systems analysis portion of the SDLC and serve as checkpoints for management reevaluation of the project. This logical and objective methodology supports an approach that has been dubbed a "creeping commitment" approach—as the project's feasibility is progressively validated after each stage, the organization's commitment in terms of resources progressively increases. If a feasibility study determines that risk is determined to be unacceptable, despite any preceding commitments, then management should not continue with it. [Ref. 25:p. 767]

In the early stages of systems analysis, such as the survey and study phase, the feasibility study cannot be completed with great detail as no technically specific solutions have been investigated. In the last phase of systems analysis, the selection phase, technical alternatives have been specified and the feasibility study becomes much more accurate. It is important for feasibility studies to be conducted—to some degree—after each phase as conclusions can change with a better understanding of current problems, business and end-user requirements, and alternative solutions. Projects that may appear to be extremely appealing at first may later turn out to be unattractive in light of increased information. It is through the process of constantly challenging the organizational, technical, and economic implications of the proposed project that IS risks are objectively assessed and minimized.

B. ORGANIZATIONAL ANALYSIS

It is no longer enough for the IS department to merely support and respond to ad hoc departmental needs. The role of IT and the corresponding IS architecture has become critically linked with the strategy and success of an organization. CEOs and CIOs need to take a look at many organizational factors prior to making any commitments on whether to downsize and prior to deciding what to downsize to. Downsizing is a process that reshapes the way organizations work and compete; top-level managers must ensure that the process will work and that the reshaped organization is in the intended form. In order to do this, several prerequisites must be fulfilled. Unless these prerequisites are met, the organization faces potentially high risks of failure. These failures may be manifested by either a failure to meet cost and performance goals, or a transition to a downsized environment that fails to meet organizational requirements. In order to avoid these risks, the top managers like the CEO and CIO must:

- Understand the business needs
- Understand the business processes
- Generate organizational support

This section will examine these top-level manager responsibilities to the organization—all essential requirements that must be met prior to committing to the downsizing of major information systems.

1. Understanding the Business Needs

The previous chapter described IT as a strategic asset. IT, however, can only be a strategic asset to the degree that it supports the business strategy. IS systems must work in concert with business strategy—one must be a reflection of the other (it should be footnoted, here, that a "business" strategy is not limited to private-sector firms; DoD units, such as ONI also need to develop a similar strategy on how best to carry out their assigned missions). Thus, as business strategy is planned and outlined, so too must IS strategy be planned and outlined. In this process, specific methodologies may be used to analyze or ensure that IS aligns itself to the business strategy. For downsizing to work, planning efforts must ensure that the downsized product is relevant to the corporate strategy. Accordingly, DoD and private corporations need to examine and understand the key ingredients that make their organization function successfully.

Analysis of critical success factors in strategic IS planning may serve as a useful first step. According to one IS strategist, "critical success factors are the limited number of areas in which satisfactory results will ensure competitive performance for the individual department or organization. Critical success factors are the few key areas where 'things must go right' for the business to flourish and the manager's goals to be attained" [Ref. 26]. Critical success factors can help a corporation analyze information systems they need to develop, maintain, or change.

The management of Federal Express, the leading overnight package carrier in the world, lists five critical success factors upon which the company's phenomenal success depends. The first of these, "(to) continuously improve quality," is integrally linked to IT. Federal express handles 1.5 million packages a day, using 430 aircraft, 31,000 delivery vans, and 91,000 employees at 1,700 locations worldwide. Federal Express' COSMOS IIB IS system, which consists of nine IBM 3090s linked to 75 dispatch centers and 24 call centers, handles the resulting 14 million transactions a day or 10 transactions per package in the system. Not only is COSMOS IIB the major on-line system for handling routing information and tracking package status, but it is also the key component in providing feedback for employees on quality factors such as on-time delivery, correct billing, calls answered on time, resolution of customer complaints, and employee satisfaction. For Federal Express, management understands that quality is an essential factor for success—and that their IS system is largely responsible for implementing that strategy. [Ref. 1:p. 72-78]

Another popular methodology (among a vast array of others) for analysis of business needs is the scenario approach. As the name implies, the scenario approach uses "what if" hypothetical situations to analyze and plan for future business needs. In these scenarios, top managers employ key variables such as the environment, trends, and events to analyze and weigh what *could* happen under different sets of circumstances. Computer based decision support systems (DSS) can be used to help quantify the results and provide objective results with which to help develop and implement the most appropriate IS architecture. [Ref. 1:p. 114] No single method is highlighted as a more appropriate than another; what all have in common, though, are systematic and objective approaches to understanding the business needs for the purpose of planning and applying the most advantageous IS architecture.

2. Understanding the Business Processes

Understanding of the business processes goes one step beyond identification of the business needs. A business need may be fulfilled without an understanding of the business processes. What can result is a system that works, but not as well as it *could* work. With an understanding of the business processes, all aspects of a system are broken down to their individual components to the point where all linkages, interdependencies, and relationships among parts are clearly understood. With this accomplished, an organization can identify and correct inefficiencies in a system. Information "barriers, bottlenecks, and blackholes" can be isolated prior to automation of the process by implementation of a new system. By taking this step back from the playing field, IS professionals avoid the mistake of applying the right technology to the wrong problem. By removing those inefficiencies in the system, the only processes that theoretically remain are those that *add value*. Streamlining of these remaining processes ensures maximum efficiency and productivity. Through this rethinking of the business processes, IS professionals identify the core value-adding processes. When this process is automated, productivity can increase dramatically instead of by token incremental improvement.

a. Business Reengineering

In their book *Reengineering the Corporation*, authors Michael Hammer and James Champy champion the idea of reinventing American companies through the process of business reengineering—"the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service, and speed" [Ref. 27:p. 32]. The four key words that the authors extract from this definition to explain this concept are fundamental, radical, dramatic, and processes.

The first key word, "fundamental," refers to the notion that companies must ask themselves basic and fundamental questions about their operations: "Why do we do what we do? And why do we do it the way we do?" [Ref. 27:p. 32]. Often, this questioning helps identify commonly held, yet obsolete and faulty, assumptions about how business is conducted.

"Radical" refers to the need for businesses to avoid timidly superficial changes in the business processes; "reengineering is about business *reinvention*—not business improvement, business enhancement, or business modification" [Ref. 27:p. 33].

When using the word "dramatic" in the definition, Hammer and Champy advocate major, non-incremental changes to the processes. More traditional approaches have called for the fine-tuning of existing methods. When called for, Hammer and Champy argue for a different approach:

> Reengineering isn't about making marginal or incremental improvements but about achieving quantum leaps in performance. If a company falls 10 percent short of where it should be, if its costs come in 10 percent too high, if its quality is 10 percent too low, if its customer service performance needs a 10 percent boost, that company does *not* need reengineering. More conventional methods, from exhorting the troops to establishing incremental quality programs, can dig a company out of a 10 percent hole. Reengineering should be brought in only when a need exists for heavy blasting. Marginal improvement requires fine-tuning; dramatic improvement demands blowing up the old and replacing it with something new. [Ref. 27:p. 33-34]

The last key word in the definition, "process," is the most important element in business reengineering, yet the most difficult conceptually for managers. Corporate managers have been bred to instinctively relate to their jobs in terms of people, their job descriptions, tasks, and organizational structures. A process is different in that it focuses not on these existing constraints, but on the collection of activities that acts on one or more inputs to create a value-added output. Hammer and Champy call for the fundamental rethinking and radical redesign of this collection of activities, not the familiar organizational entities. In traditional attempts to cope with crises, corporations have typically selected from a menu of three choices to resolve their dilemma: (1) bring in new people, (2) automate the task with bigger and faster machines, or (3) restructure the organizational chart. These options usually result in only superficial and marginal improvements. By reengineering the processes, a new option is added to the menu, with potentially dramatic and radical improvement. [Ref. 27]

As suggested in the previous chapter, IT can play a key role in business reengineering as an enabling force. According to Hammer and Champy, however, IT's key role is all too often miscast and results in exacerbating the existing situation. Automation of *existing* processes, Hammer and Champy insist, is *not* reengineering.

> ...to paraphrase what is often said about money and government, merely throwing computers at an existing business problem does not cause it to be reengineered. In fact, the *misuse* of technology can block reengineering altogether by reinforcing old ways of thinking and old behavior patterns. [Ref. 27]

Using IT to challenge old assumptions, alter outmoded and inefficient operating methods, and to break traditional rules *is* what reengineering is all about. Utilization of Electronic Data Interchange to break traditional organizational boundaries, exploitation of database technology to share information simultaneously, and harnessing the power of microprocessor to bring mainframe computing power to the desktop are all examples of freeing an organization from constraining assumptions. What IT enables is the notion of *discontinuous thinking*—"identifying and abandoning the outdated rules and fundamental assumptions that underlie current business operations" [Ref. 27:p.3].

b. The Total Quality Movement

Hammer's and Champy's program of business reengineering shares some common ground with some familiar business improvement programs, but differs radically from others. The Total Quality Management (TQM) program and the U.S. Navy's Total Quality Leadership (TQL) initiatives, for example, share a similar theme with business reengineering in that both recognize the importance of studying the business processes and the primacy of the customer. Where TQM and TQL differ, though, is that these programs seek to gain improvement of business process through continuous, incremental improvement rather that business reengineering's radical and dramatic remedy.

> Quality programs work within the framework of a company's existing processes and seek to enhance them by means of what the Japanese call *kaizen*, or continuous in-

cremental improvement. The aim is to do what we already do, only to do it better. [Ref. 27: p.49].

Manifestations of the total quality movement are seen in various management and organizational development techniques designed to improve employee productivity. Dramatic success stories in management, such as the Japanese utilization of W. Edward Deming's TQM program to help revitalize the Japanese economy, have fostered a greater appreciation for innovative management principles in the American economy. These new principles stress quality, process redesign, employee involvement, self-managing work-groups, teamwork, and job redesign. Whether the methods for approaching the analysis of an organization have been business reengineering, TQM, or the sociotechnical system (STS) approach, the pervasive trend has been a closer examination of the more elusive, non-traditional aspects of organizations with an emphasis on the business processes. Peter Senge, director of the Systems Thinking and Organizational Learning Program at MIT's Sloan School of Management, encourages such thinking in his recent book, *The Fifth Discipline: The Art and Practice of the Learning Organization*, in which he prompts organizations to become *learning* organizations in order to survive [Ref. 28].

As evidenced through numerous corporate case studies, IT can act as a catalyst for Hammer's and Champy's version of business reengineering and other programs of process improvement. IT has characteristically been the tool through which traditional assumptions have been broken and new ways of operating have become possible. The opportunities afforded by the new technologies available in a downsized environment can translate into new processes for doing business, new windows of opportunity. Nonetheless, when assessing the risks inherent in downsizing, it is inherently important to understand the processes of the business. That is what this section has tried to emphasize. Whether or not a business ultimately undertakes a downsizing proposition, a critical understanding of the processes can ensure a better evaluation of the possibly revolutionizing benefits of downsizing or the overly daunting pitfalls.

3. Generating Organizational Support

The greatest risks with downsizing may not be associated with the proper analysis of business process, a thorough performance analysis, or an accurate cost-benefit justification. Instead, the downsizing issues that may pose the greatest obstacles to progress are related to human factors. Downsizing can have generate organizational changes that can have a profound impact on an organization's operating procedures and corporate culture. The resulting fear factor can create widespread and unpredictable consequences that can affect the success of the downsizing process. Before proceeding too quickly with any downsizing plans, it is an absolute necessity to ensure that top corporate officials have "bought off" on the idea. Unless sustained top management commitment is forthcoming to help initiate the process and overcome a resistance to change, the success of downsizing is at risk.

a. Top Management Support

Downsizing is as much a political issue as a technical issue. Top management commitment is crucial in light of a potentially difficult migration process full of volatile cultural and social upheaval. External factors such as underestimation of development or training costs can shortchange the process. Internal conflict due to potential job losses can delay and even sabotage progress. In face of unanticipated pitfalls, lack of absolute conviction and support to guide the downsizing initiative will exacerbate a deteriorating situation. Lack of executive support to provide constant leadership and guidance within a strategic framework will ensure failure.

Despite the criticality of obtaining top management support in downsizing information systems, the task may be more challenging than one might expect. A study done by the Massachusetts Institute of Technology's (MIT) Sloan School revealed surprising attitudes towards information systems on the part of CEOs. [Ref. 1:p. 54-56] The study involved two-hour interviews with CEOs from 84 organizations within ten different industries and focussed around the CEOs' perceived value and expectations from IT investments. The results were disappointing for IS departments. **Fifty-two percent** of the CEOs said that they were *too unknowledgable* about IT to direct their investments. The **other 48 percent** (see Figure 11) of the CEOs had varying attitudes about IT that reflected various degrees of confidence in informations systems themselves and the IS departments. What the study reveals is greater CEO confidence in the value of IS technology (38 percent) than confidence in the personnel (20 percent). Furthermore, only 12 percent of the CEOs had high confidence in both the technology and the people. Fiɡ ɹe 11 illustrates why obtaining top management support may be more difficult than anticipated.

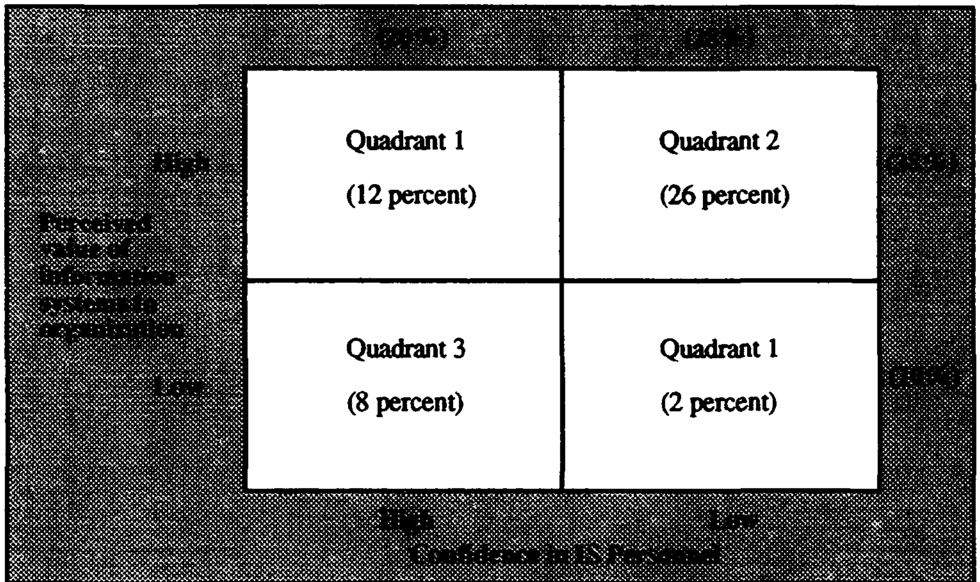


Figure 11: CIO Perceptions of IS Systems and Personnel [Ref. 1: p. 55]

In order to overcome possible foot-dragging by CEOs, it becomes the responsibility of the CIO to provide the leadership necessary to generate top-management confidence in IS technology and personnel. If it does not already exist, establishing confidence in IS will not be an overnight process. Because of the career-threatening high stakes involved in IT, the acronym CIO has sometimes facetiously been said to stand for "Career Is Over." Instead of fostering that negativism, Michael Hammer suggests that it should stand for "Chief Innovation Officer," with the IT executive using the position to take charge "as a leader of a team that identifies new opportunities and implements new ways of doing business" [Ref. 3:p. 11]. Accordingly, the CIO must establish credibility by showing he or she understands business needs and processes and demonstrating how the IS department can help increase the technological maturity of an organization through appropriate technologies. The CIO's ability to convincingly demonstrate and sell a vision for new technologies to the CEO is the key hurdle for any major IS project. Getting the CIO and CEO to function as a team is a major prerequisite to further progress.

b. Managing Resistance to Change

With a united and committed management, the next step is to solicit the entire organization's dedication. Downsizing information systems can have a profound organizational impact that can trigger many unanticipated responses. If the downsizing project involves a major move off the mainframe to a PC LAN, for example, there will be an inherent shift in the need for hardware and software skills. Those that may have spent their entire careers within the glass house are likely to be ill-equipped to adapt to the requirements of the new downsized environment. Little incentive exists for a COBOL programmer to assist in the downsizing process when the individual realizes he or she is essentially helping to make their position obsolete. In fact, much incentive exists for such an individual to discourage a downsizing effort and perpetuate the status quo.

Unavoidably, downsizing will require change—and inevitably, change will provoke resistance. Reactions to change will vary depending on the circumstances, but organizations commonly respond to change in four negative ways: (1) feelings of incompetency, powerlessness, and being unneeded among personnel, (2) an atmosphere of confusion and unpredictability throughout the organization, (3) increased instances of conflict, and (4) a sense of loss. [Ref. 29:p. 397]

Unless proactive steps are taken to counter these negative reactions, the forces of resistance can work to undermine any downsizing initiatives. Thus, the CEO and CIO must acknowledge these predictable responses and work in concert to match each, in turn, with a countermeasure. Four strategies should be considered to deal with the changes: (1) establishment of training and support for the employees, (2) realignment of formal roles and relationships to coincide with change, (3) establishment of arenas for discussion and problem-solving, and (4) development of transition rituals to herald the "new" organization [Ref. 29:p. 397]. Planning and implementation of these countermeasures to resistance can help dissolve snowballing forces of resistance from gaining momentum and thus minimize the risks of downsizing (see Figure 12). Nonetheless, the resistance to change and associated emotions brought on by downsizing are an unpredictable risk. Consideration and understanding of the human element are essential in an effective risk analysis of downsizing.

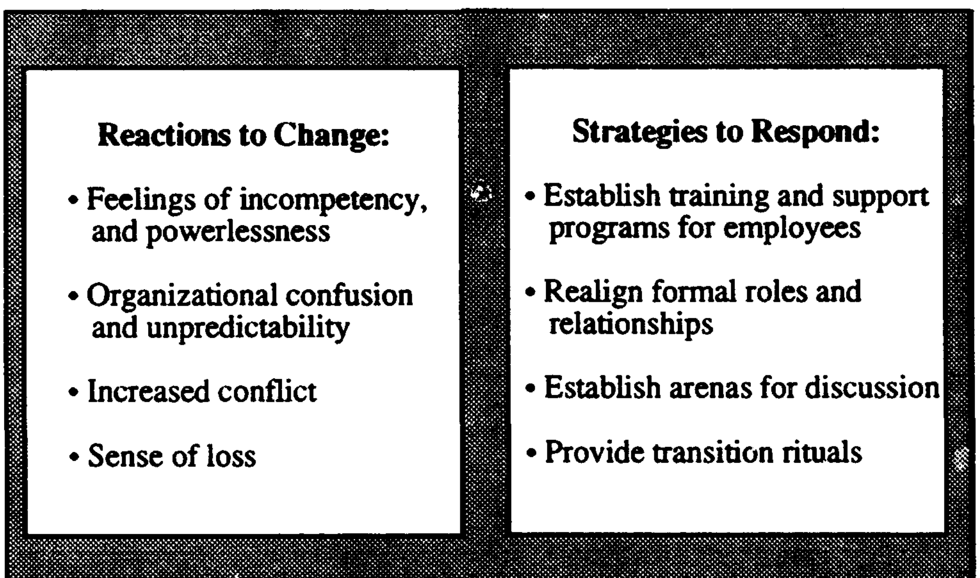


Figure 12: Managing Resistance to Change

C. TECHNICAL ANALYSIS

The process of properly assessing risk requires examination of different management and technical perspectives. The previous section focused on organizational aspects of risk; this one looks at the technical aspects. An organizationally sound project does not necessarily mean that technical criteria are being fulfilled. CIOs must assess the feasibility of a downsizing project by also asking and answering several key technical questions:

- Is the proposed technological solution practical? [Ref. 26:p. 773]
- Will it meet all of our current and desired performance needs?

1. Is the Technological Solution Practical?

Although today's rapidly advancing technology allows for the possibility of many technical solutions, this does not necessarily imply that the solution is a practical one. Qualifying as *practical* requires testing the proposed solution against such issues as system maturity and the track record of technology. A practical system also implies that it is readily achievable. The complexity generated by the heterogeneity in today's computing environment does not necessarily mean that a distributed system will meet that requirement. Thus, the potential lack of standards and interoperability need to be examined. Finally, a solution cannot be practical if there is not adequate expertise or time available to develop and implement the proposed solution.

a. System Maturity

Many argue that in contrast to the mainframe's legacy of stability, the PC and workstation have just entered into the big league of corporate computing and are still establishing and proving themselves as a technologically viable solution. A risk-averse organization, wary of vendors hawking advantages of distributed computing in a LAN environment, may look at 30 years of mainframe experience in managing critical systems in large organizations and choose to bet on proven technology. Undeniably, the mainframe boasts an impressive track record of successes (as well as an embarrassing number of failures). Perhaps this systems maturity and adaptation to its environment explains a recent study conducted by the Business Research Group of Newton, Massachusetts, that estimated that 75 percent of all computing tasks are still being performed on the mainframe [Ref. 30:p. 36].

One explanation of the reluctance to downsize to smaller systems follows:

> Savings can be achieved by moving applications to distributed networks or smaller host processors, but such economies of scale alone are seldom motivation enough to abandon mainframes... Many applications run on mainframes because they are of such enormous financial value or critical importance to the organization itself. As a result, information systems directors are reticent to trust such "mission-critical" applications and data outside of the mainframe environment; they simply feel uneasy about losing the stability and reliability upon which they have depended for so long. [Ref. 30:p. 36]

Others agree that the mainframe is a mature system, but footnote that it is 30 years *too mature*—a modern day dinosaur by computing standards. This camp believes that the technology available on the desktop *is* mature and capable enough to meet their requirements and if they do not invest in the future now, there will be no future for them. The argument is that they cannot *not* afford to downsize unless they are willing to lose any competitive advantage and possibly go out of business. To them, the debate is not whether to downsize, but when and what to downsize.

b. Proprietary vs. Open Systems

Another key issue that must be technically analyzed when selecting or approving a system is how proprietary or *open* the target architecture is. The proprietary system, a label that has most often been attached to the mainframe (although there have been significant recent strides toward more open systems), has its obvious disadvantages of single vendor lock-in and high cost by not allowing interoperability with other commercial components. *Open systems* try to achieve a state where all components are interoperable. Open systems are defined as:

> An approach to building information-processing systems using hardware, software, and networking components that comply with industry-accepted standards such as OSI (Open Systems Interconnect) or POSIX. [Ref. 6:p. 353]

What this requires is a comprehensive set of standards that will allow construction of integrated systems and will provide interoperability among systems (i.e., exchange of data) and portability of applications. For the IS department, an environment of open computing systems would not only generate the potential for considerable cost savings (driven by intense competition and the resulting economies of scale), but it would also permit:

- IS departments to gain independence from any specific IT vendor.
- IS integration to be achievable across heterogeneous environments.
- The opportunity to choose the *best of breed* in any technology area. [Ref. 20:p. 1]

Open systems may be practical, but the more important underlying question is whether these systems are achievable. Reliably running and maintaining different system elements running on different sets of computers is an inherently difficult accomplishment. Chief among the risks of a heterogeneous distributed environment are the accompanying technical complexity and the lack of any true standard to mitigate the effects of that complexity and heterogeneity. Gartner Group believes the goal of complete interoperability has created an "architectural crisis" or sorts:

> IT moved from an era of *manageable, homogeneous computing on monolithic systems* in 1980, to an era of *unmanageable, heterogeneous, networked systems by 1990*. In 1980, after many years of practice, IS understood how to implement applications systems, buy from vendors and deal with end users. By 1990, virtually all aspects of information systems operating procedures were being overthrown, primarily because of the introduction of PCs and distributed, heterogeneous systems. Computing architectures have spun out of control. Local Area Networks (LANs) have experienced runaway growth, and desktops and servers of all sizes and persuasions have

appeared like weeds. Programming standards have eroded and the notion of non-redundant, consistent, high-quality data is on the compost heap. [Ref. 7:p. 1]

The range of options created by the technological flexibility and the response to the demand for open systems has multiplied the complexity of architectural choices and the decision-making process and changed the procurement process. The quest for an open environment has become a system user's dream, but a system integrator's nightmare:

> Consider that in an open systems environment there are four or five major platforms to choose from, four or more database management system providers, five or more communications alternatives, five or six productivity environments and so on. It is easy to conclude that there can be 6,000 technical combinations possible for such an open environment. (I have actually seen estimates of millions of combinations.) [Ref. 10:p. 37]

A truly open system would require universal compliance to one set of standards; this is where the problems arise (in fact, only proprietary systems offer this option). With much invested in already developed proprietary standards, software developers representing major computing firms are trying to convince the market to adopt their standard to protect their investment and secure an economically advantageous foothold. The UNIX operating system, for example, regarded by some as running the most open operating system available, has been fragmented by corporate politics and maneuvering. Being written in C enables UNIX to be a very portable operating system; however, the many vendor modifications to it as part of proprietary infrastructures have resulted in the proliferation of different versions. Recent efforts by major UNIX vendors such as IBM, Hewlett-Packard, and Sun Microsystems have been made to try to unify the operating systems as part of a Common Open Software Environment (COSE). The verdict on this eleventh hour attempt to unify UNIX—coming only after the imminent introduction of Microsoft's Windows NT—is still out.

This corporate jockeying for market position in the world of developing standards is not taking place only in the arena of operating systems. The battle of competing standards is being waged full force in the area of networking protocols, user interface environments (UIE), distributed data access, transaction processing (TP) monitors, applications languages, and repositories. The goal of all this is to provide for the seamless interface in a distributed environment where end-users will be able to run applications without awareness of the system software, hardware platform, or network protocols being used. To a large measure, much of this seamlessness is to be accomplished through the use of *middleware* (see middleware discussion in Chapter III), the comprehensive collection of application programming interfaces that will veil the true system complexity and heterogeneity. Middleware vendors and products such as IBM's System Application Architecture, Open Software Foundations DCE and DME, and Microsoft's ODBC and WOSA hope to compete for this growing market.

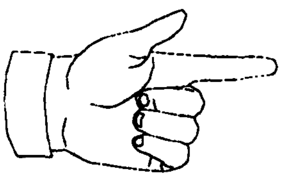
Thus, the complexity and confusion stemming from the movement to open systems appear to be other manifestations of this architectural crisis that has been generated by the prodigious growth of *distributed heterogeneous computing environment*. If the downsized target architecture is such a non-proprietary and heterogeneous system, the risks associated with open systems—an unmanageable complexity or commitment to an inappropriate or fading standard—must be considered. Furthermore, waiting for resolution of differing standards may be counterproductive; comprehensive and clearly dominant standards may never emerge. The Gartner Group terms "openness in the absolute" as "nonsense" and in the long run predicts a coexistence of multiple standards with interoperability between them [Ref. 31:p. 4].

c. Technical Expertise and Time Constraints

Finally, two important and somewhat obvious practical issues involve the technical expertise of personnel and schedule requirements. Often disregarded or optimistically estimated, these two constraints can foil a downsizing project and a potential business opportunity. The technology exists to implement a wide array of potential downsizing solutions; only well trained personnel can properly choose the most appropriate of the seemingly infinite possibilities. Success in complying with a deadline, while still providing a quality solution, will also be directly dependent on the technical abilities of personnel.

Downsizing from a mainframe to a distributed computing environment requires shifting to a new set of technical skills. There is not so much a lack of personnel, as the right kind of personnel. The downsized architectures come equipped with a different set of tools—many more efficient and productive than their mainframe counterparts. The technological advances and advantages made available by OOP, 4GLs, and GUI interfaces, for example, are too impressive to forgo. Unfortunately, the mainframe is not usually the platform of choice for implementing these tools; as a result, the mainframe bureaucracy is increasingly becoming "deskilled" in new developments. Some IS professionals might argue that their departments cannot afford to *not* advance and learn with these new developments. Corporate conferences on downsizing often compare the needed skills of today with those of yesterday with illustrations such as Figure 13.

CHANGING SKILLS OF IS DEPARTMENT



Yesterday:

- Cobol
- Conventional databases
- Computer science degree

Today:

- C, CASE, OOP
- SQL and client/server
- Business and technical degrees

Figure 13: The Changing Skills of IS [Ref. 32:p. 214]

Strategies to cope with a shortage of technical skills vary from retraining, to the hiring of consultants, to outsourcing. The hiring of outside system integrators and consultants has often provided efficient and effective relief to organizations with an overwhelmed and underskilled staff facing downsizing problems. Increasingly, however, retraining of internal personnel has been seen as the long term solution to a shortage of technical skills—in light of fears that the consulting option "may leave you holding the bag once the new platform is up and running" [Ref. 33:p. 82]. A more conservative strategy, designed to minimize risk, is to conduct comprehensive retraining *with* the consulting firm *while* downsizing.

2. Will the Downsized System Meet Performance Requirements?

When it comes to downsizing, two distinctive camps of IS professionals are separated by a wide spectrum of performance issues. According to one camp, downsized systems offer viable alternatives to the mainframe and promise similar performance with huge savings. The opportunity to migrate to smaller systems, that appear to offer similar speed and storage capabilities at a fraction of the mainframe costs, is too attractive for them to resist. They support a mass exodus to the distributed computing environment. The other camp staunchly defends the centralized environment and its track record for handling complex, mission-critical applications. Expressing doubt over the technical immaturity of smaller systems, they believe that only the mainframe can provide the full range of required functionality within certain performance parameters.

The next few sections will examine both sides of the debate. In the context of this thesis, "performance" connotes characteristics such as flexibility and scalability, processing speed (MIPS rates), throughput, integrity and security, and availability.

a. Flexibility and Scalability

Computing systems can evolve and grow in two different directions, either vertically or horizontally. A vertically expanding system is one that grows by increasing the resources of a single computing node with the addition of more memory, more storage capability, more terminals, or more processors. In contrast, a horizontally expanding system grows by adding more nodes rather than increasing the resources at a single element.

In the past, the traditional mainframe computing systems have sometimes only been allowed to expand vertically. With rapid technological advancements in processors and increasing corporate appetites for more processing power, typical host systems became outdated and inadequate after five productive years. At this critical junction, a corporation that may only have needed a fractional increase in computing power had limited options to make small increments to capacity (e.g., by adding more memory, more I/O channels, etc.). At some point, they needed to buy entirely new mainframes that may have dramatically exceed new requirements. Historically, it has also been the case in traditional product lines that the same operating systems and application development environment (ADE) have not spanned the range of computing architectures. Thus, a corporation that needed to upgrade from a mid-range computer to a larger mainframe was forced to go through the major and disruptive evolution of migrating the software to the new system and go through a new learning process for all IS personnel. This type of vertical growth has three major limitations:

- potential to reach systems limit
- inflexibility in configuration
- potential change in software environments [Ref. 34:p. 100]

To be sure, the degree of mainframe scalability across its systems, interoperability among systems, and consistency in its software and ADE has grown proportionally to the increased market demand for such features. And, to a considerable extent, these mainframes characteristics have evolved dramatically since its early beginnings. Systems scalability and flexibility, however, tends to be much simpler in a downsized environment. In terms of granularity (the size of the step that must be taken to increase in functionality such as processing power or storage capability) and the corresponding price/performance ratio, the popular consensus is that the mainframe cannot compete with a workstation or PC in a distributed environment. The granularity of the system is also complemented by configuration flexibility, offering the capacity to increase resources in different locations to accommodate different IS needs of the organization.

b. Processing Speed

One measurement that is often used to compare technical capabilities relative to costs has been the price to performance ratio. Millions of instructions per second (MIPS) rates tend to be the most widely used measure of performance, and are believed to represent the processing power accomplished by one or more processors.This performance measure is both so commonly used and misunderstood that some suggest a more appropriate breakout of the acronym is “meaningless indicator of processing speed.”

Firstly, measurements of MIPS differ between different architectures; RISC MIPS, IBM S/390 MIPS and Intel MIPS, for example, cannot fairly be judged against one another as all use different sets of instructions. Furthermore, it is important to note that MIPS is a measure of instructions executed per unit time, and *not throughput*. The MIPS rate required varies according to a number of factors. Computing machines that are comprised of operating systems, database, and communications systems may be required to execute more MIPS and utilize more processing time to provide the necessary level of functionality than a system that does not have all these components. Additionally, some application software and compilers are written more efficiently than others and require a fewer number of instructions to be executed.

The phenomenal increase in the MIPS capabilities of smaller machines has enabled a network of workstations in a downsized environment to appear as a much more attractive computing alternative than the mainframe, especially when comparing the price/performance ratios. For the majority of desktop applications, however, MIPS issues are largely irrelevant in light of the minimal CPU execution time as compared to the I/O wait time. In many instances, MIPS rates are a relatively insignificant feature and become an important factor only when transactions are CPU-intensive (such as in scientific applications) due to time-consuming processing requirements. Thus, MIPS rates should only become a significant performance requirement and a potential technical risk when an application is processing intensive and the CPU becomes the source of a bottleneck.

The Gartner Group outlines a similar argument when comparing mainframe to PC MIPS:

> If you go to a typical mainframe shop and count the on-line DASD and divide it by the aggregate processing power, you would find that there are between 3,000 and 5,000 megabytes of data per MIPS. In a typical workstation, the ratio is more like 20 to 50 megabytes per MIPS. That is a two orders of magnitude difference. By that ratio, mainframes are data-rich/MIPS-poor platforms, and comparably speaking, workstations are MIPS-rich/data-poor platforms. Which is why you tune for MIPS on a mainframe because it is the most constrained resource... We would never propose to use dollars per MIPS for making a platform choice. It is an interesting metric within a family of processors, but is not interesting between different types of processors. [Ref. 7:p. 63]

c. Throughput

The mainframe's core strength is often seen as its ability to efficiently manage and move large volumes of data. Technically speaking this translates to low *response times* and correspondingly high *throughput*. The hardware, operating systems, and supporting software have all been optimized to support this data management capability (as discussed in Chapter II). The enormous bandwidth and the ability to manage multiple complex tasks simultaneously have become mission-critical elements for some corporations; these capabilities are often needed to support bandwidth-intense applications that utilize multimedia, graphics, video, image, and voice. Graphical user interfaces, 4GLs, and flexibility—strengths of the desktop computing architecture—may not be pertinent in many cases. According to one expert:

> The mainframe's strength is maintaining order in what could all too easily become chaos in a totally distributed environment: assigning order of admission, changing data, logging in updates, and being prepared for disaster recovery. The mainframe is enormously efficient at anticipating an I/O gap and dealing with another job simultaneously to optimize processor performance and minimize the wait-state. Parallelism, bandwidth and well-developed software applications give it incomparable throughput capability. [Ref. 35:p. 34]

One of the areas that the mainframe hardware and software have been groomed for efficiency over the last three decades has been on-line transaction processing (OLTP). Because MVS-compatible programs and applications have established a proven track record in protecting data integrity and guaranteeing reasonable response times, those systems currently manage the vast majority of data for large enterprises such as banks, airlines, and insurance companies. Corporate customers continue to rely on the mainframe as a platform of stability for mission-critical OLTP. Admittedly, speed, storage capacity, and cost factors are important. Yet, what is the most cost-effective way to work? An IS structured around the mainframe has repeatedly shown itself to be a system capable of meeting incessant demands of unpredictable complexity in the face of critical deadlines. Many systems dependent on OLTP demand the right mix of MIPS, I/O (input/output), and throughput to maintain their competitive advantage. Many companies prefer to rely on the robust performance of the mainframe rather than venture into questionable risks associated with a downsizing project.

For years, the workstation's progress in the OLTP market has been constrained by its limited I/O capabilities. In commercial record-handling applications, 20 to 30 disk accesses may be required for each transaction of medium complexity; a processing rate of 10 such transactions per second thus requires 200-300 accesses per second. Though this rate can now be handled by minicomputers and workstations, the technology has only recently become readily available.

Despite advances in the I/O capabilities of workstations, the several hundred disk accesses per second achievable by these servers do not match the mainframe's capabilities. The 10,000-15,000 I/Os per second that are manageable on a large mainframe make it the platform of choice for corporate applications that require such demanding transaction rates. [Ref. 4:p. 28 & 67]

Thus, though minicomputers and workstations that have not been able to match top of the line mainframe OLTP performance in these areas, these smaller systems may satisfy less demanding OLTP applications with moderate transaction rate requirements. One article entitled "PCs Break Into the OLTP Ranks" explains the market change:

> That (OLTP) picture is beginning to change. Powerful new PC hardware and operating systems provide the performance, security, multiprocessing and multitasking capabilities usually associated with much larger systems, making downsized corporate backbone applications possible. And that's attracted vendors of transaction-processing (TP) monitors. These are the systems that furnish the security, data management, communications and development tools needed to build and run OLTP apps. [Ref. 35:p. 32]

The author of this article goes on to support his claim by providing examples of three different companies that have developed UNIX TP monitors that will support the ability for Macintoshes, PCs, and workstations running Windows, OS/2, and UNIX to function as full-featured clients. Additionally, fully downsized versions of IBM's Customer Information Control System (CICS)—the most pervasive and acclaimed of all TP monitors—are now available on OS/2, IBM's version of UNIX (AIX), and Windows NT. The administrative computing department at Tulane University in New Orleans claims to have saved $135,000 per year in maintenance costs alone by moving only a portion of its CICS applications down to 486-based PCs and RS/6000 workstations running OS/2 and AIX. [Ref. 35:p. 32]

d. *Integrity and Security*

Decisions to downsize information systems have sometimes occurred based on the myopic focus on potential cost savings while neglecting other important concerns such as integrity and security. Because admissions of security loopholes are shunned, documentation of case histories lamenting costly security oversight is not readily available. Nevertheless, the lack of a guarantee of proper security is a major cause for concern when downsizing. One major belief is that mainframe-strength security systems are simply not as readily available—or as robust—in a distributed environment.

The seriousness of this perceived problem is compounded by the fact that it is extraordinarily difficult to manage security in an environment that has constantly changing hardware, operating systems, and databases. It appears that the features that are so often highlighted as an important advantages of distributed systems—cost savings and flexibility—come at the expense of security. "When companies cannot afford all the security involved, they just don't talk about it" commented a Database Associates professional in Morgan Hill, California. "People don't seem to worry about it too much. They keep their fingers crossed that it won't cause major problems." [Ref. 36:p. 25]

While some organizations are not willing to put much stock in security at the LAN level, others have already made their commitments to downsizing and are resolving the security issues through a variety of methods. One of these methods is to firewall a LAN from external access. Such an approach was taken by Burlington Coat Factory Warehouse in Lebanon, N.H. when it junked its mainframe in February 1992 and committed itself to an open and distributed system. The firm purchased a central security system form Sequent Computer Systems, Inc. in Beaverton, Oregon for six large Unix-based servers that connected several hundred workstations and several thousand PCs. The inevitable trade-off they faced was the decreasing access with an increasing level of security.

Even so, many corporations feel that downsized systems can meet or exceed mainframe equivalent security. [Ref. 36:p. 25] In fact, the Gartner Group also takes this position in a series about downsizing with the following assertion:

> Downsizing does not necessarily compromise a system's general trustworthiness... Although ease of access, a secondary objective of downsizing, could cause exposure in security management, the lack of security is not a technical issue. Many systems, particularly in the midrange, have been qualified at higher Department of Defense security levels than mainframes. [Ref. 37:p. 4]

e. Availability

Unpredictable system outages that prevent access to mission-critical applications can be very costly to an organization. Continuous availability, the expectancy of reliable service without system outages, has almost become a built-in feature of the mainframe environment and has contributed to its reputation of stability. Back-up components automatically take over in case of failure. The centralized nature of the mainframe and the concentration of processing resources naturally supports such availability because of the fewer possible points of failure.

The issue of availability in distributes systems, however, is a little more complex. At first it would appear that a commitment to distributed computing may require a concomitant acceptance of reduced availability due to the many possible points of failure. By its very nature, however, a distributed system provides redundancy. Even if a local processor fails, the entire system, in general, will not fail.

Although the mainframe has generally enjoyed a better reputation for providing a high degree of availability, the distributed computing environment is making up lost ground rapidly with advances in fault-tolerant technology. Fault tolerant technology creates redundancy within a distributed architecture by duplicating every major component within one module. If a component fails, a diagnostic link will immediately inform the service center and automatically enable a substitute component while the faulty component is repaired within 24 hours. As it becomes prohibitively expensive to duplicate all components, redundancy only at strategic locations becomes a reasonable compromise. Successful implementations of this strategy have been implemented with favorable results. Often the fault-tolerant system is used as a front-end to a mainframe in a client/server type environment.

D. COST/BENEFIT ANALYSIS

The purpose of the cost/benefit analysis is to summarize all of the estimated costs associated with a downsizing project in sufficient detail to give top decision makers the ability to decide whether or not to proceed with the process. Costs and benefits should be incrementally more accurate with each succeeding phase of the SDLC. Traditional steps for performing a cost/benefit analysis require estimation of a variety of factors:

- cost of operating the current system
- cost of the downsized system or proposed system
- costs associated with all of the phases of the SDLC
- description of the intangible costs and benefits
- basis for estimating how the above costs and benefits will change in the next few years due to such factors as changing economic and business trends
- quantification of the risks for proceeding or discontinuing the project [Ref. 6:p. 71]

Putting these guidelines in the context of a downsizing project, the return on investment may be determined by *comparing the conversion costs and the costs of operating in the new environment* ***against*** *the cost of continuing to operate in the current environment—all over a set period of time (e.g., five years).* This section will analyze potential conversion costs and highlight the often subtle costs of operating in a distributed environment. Because not all downsizing-related costs are readily identifiable, the last sub-section will examine some intangible cost considerations.

1. Conversion Costs

The initial transition costs in a downsizing program can vary tremendously according to the scope of the project and the migration strategy. For example, if a corporation's downsizing strategy is simply to downsize one non-mission-critical application with a "surround and integrate" migration approach (see Chapter III), then the conversion costs can be relatively small. A "surround and integrate" migration strategy leaves the application relatively intact on the host mainframe and accessory applications are added as needed (typically for a GUI interface). If, however, the corporate migration strategy is to "kill" the mainframe and to off-load all applications, the transition costs will be dramatically higher.

The conversion costs fall within two categories: the cost of migrating the application and the costs of new hardware. The cost of migrating the application will depend on the migration strategy. Rewriting the code for a mission-critical application, for example, is obviously tremendously more expensive and time-consuming than buying ready to use off-the-shelf software. IS professionals need to analyze the trade-offs between costs and performance factors when deciding on a code migration strategy.

Similarly, the conversion costs associated with the purchase of new hardware will also vary significantly according to the migration strategy. A "surround and integrate" migration might entail merely replacing "dumb" 3270 terminals with intelligent workstations while keeping the bulk of the processing on the mainframe. Such a move would certainly not result in as many initial hardware expenses as a migration strategy that would scrap the mainframe. The Gartner Group estimates that:

> ...(when) simply moving applications from one platform to another, the break-even point usually falls between two and four years. However, if companies exercise the "opportunity option" to move to a more open environment or toward distributed or client/server computing, initial costs could be substantially higher and thus delay reaching the break-even point by one or two years. [Ref. 38:p. 9]

2. Operating Costs

Probably a more difficult challenge than estimating conversion costs is the task of estimating operating costs in the new environment. Many times, downsizing decisions are based on the attractive hardware price/performance ratios offered by the current desktop environment. Decisions often do not account for other quantitative and qualitative factors that are an integral part of the day-to-day operations in a distributed environment. Contrasting mainframes at $50,000 per MIPS to a UNIX machine at $500 per MIPS, for example, can be a very misleading comparison. The processing power does not take into account a myriad of the other technical and non-technical issues inherent in distributed systems.

Estimating costs of distributed systems requires looking at the cost of not just the machines themselves, but also the significant overhead that accompanies networked architectures. Costs grow as the complexity of networks grow due to factors such as increased heterogeneity, the additional layers of software (e.g., middleware) and expertise necessary to support such diversity. Identification and estimation of cost elements such as data management capabilities, labor costs, security, maintenance, ease of use, and business responsiveness—to name a few—are also crucial to completion of an accurate cost/benefit analysis, but are often downplayed.

a. Life-Cycle Cost of a Downsized System

One of the fundamental reasons that is repeatedly cited for the downsizing trend is that traditional mainframes are simply too expensive compared to networked workstations, PCs, and other downsized alternatives that are currently available in the market. The economies of scale that once made the mainframe the most cost-effective computing solution is perceived to have been inverted. The focus, however, has primarily been on the appealing hardware and software costs of downsized systems and has tended to ignore full life-cycle costs of downsized systems.

(1) Hardware costs. Hardware components represent perhaps the most visible and tangible costs. Overhead transparencies, similar to the one in the Table 4, frequently illuminate executive boardrooms to illustrate price differences and trends of initial hardware costs.

Table 4: PRICES: MAINFRAME AND WORKSTATION [Ref. 9]

Cost Per	Mainframe	Workstation
MIPS	\$100,000–\$150,000	typically less than \$500
Megabyte Memory	\$2000–\$4,000	typically less than \$100
Megabyte Disk	\$5–\$10	\$1-\$2

While these price/performance ratios of desktop systems are impressive, desktop hardware components are becoming obsolete at a faster rate than mainframe technology. Replacing hundreds or thousands of desktop computers in very large organizations every few years can become a very expensive proposition. Even with the frequent replacements of workstations, the hardware costs of desktop systems really represent only a narrow slice of the total IS budget. While Table 4 illustrates the current unit costs, Table 5 shows the historical price and performance trends for the workstation, minicomputer and mainframe. Again, though, such tables generally tend to illustrate only those aspects of computing systems that are discrete and readily quantifiable. End-users tend to be more impressed by large numbers representing speed and storage capabilities than with qualitative explanations of security and reliability. Thus, while Table 5 dramatically depicts the industry's downsizing direction and potential, it does an inadequate job in illustrating other important attributes of both PC and mainframe systems and their full life cycle costs.

Table 5: PRICE AND PERFORMANCE TRENDS [Ref. 9]

	Workstation **1993**	Minicomputer **1981**	Mainframe **1971**
MIPS	50	2.0	1.0
Off-line Storage (MB)	2000	100	20
Operating System	multi-user, multi-processing	multi-user single processing	single-user single processing
Internal Memory (MB)	32	8	2
Task Capacity (tasks)	16	8	4
Cost	$5,000	$100,000	$1,000,000

(2) Software costs. Cost comparisons of development and maintenance of mainframe versus desktop software have highlighted significant cost differences that are commonly perceived to favor smaller systems. The Consolidated Insurance Group, prior to downsizing to a PC LAN, utilized a $200,000 mainframe-based, general-ledger program to maintain its accounts. After downsizing, the insurance company switched to a $595 off-the-shelf program to perform the same functions—a 300 to 1 price advantage over the mainframe version [Ref. 6:p. 5]. Other comparisons, such as the following observation, have noted price disparities:

> In one-on-one comparisons of packaged software between mainframes or minis and PCs, the price comparison is so vast as to be almost ludicrous. Typical PC packages cost from $50 to $500; typical mini and mainframe packages cost from $5000 to $50,000. The 100-to-1 advantage in cost is a good rule of thumb. [Ref. 6:p. 57]

There is little dispute over the cost advantages of a single software package for a desktop environment over one built for a mainframe system. Organizations, however, need to make one additional consideration. How many individual software packages do they need to buy to support hundreds or thousands of networked desktops? A 100-to-1 cost advantage quickly disappears when large quantities of software packages with documentation are purchased to meet corporate-wide requirements. Furthermore, because of the shorter life-cycle of desktop software, organizations may find themselves spending money to support periodic upgrades to new and improved versions. Acquisition of site licenses can alleviate, but not remedy, this situation.

(3) Full Life-Cycle Costs. Convincing price comparisons of hardware and software costs are often used to help justify moving to smaller systems throughout industry journals and trade shows. The focus tends to be on hardware and software—*capital costs*—not the entire life cycle. One study performed by the Gartner Group helps refute the importance of capital costs. In it, the Gartner Group estimated the average five-year life cycle of an average PC in a corporate environment to be $40,124, with capital-related costs accounting for *only 15%* of the total life-cycle costs. The other 85% percent of the costs were personnel costs, related to administrative and technical support as well as end-user operations. Figure14 illustrates these findings. [Ref. 37:p. 6-7]

Total Cost	**$40,124**	**(100%)**
• Capital-related	$5,886	(15%)
• Labor-related	$34,238	(85%)
— Administration	$5,505	(14%)
— Technical Support	$6,040	(15%)
— End-user Operations	$22,693	(56%)

Figure 14: Five-Year Life Cycle Cost of a PC [Ref. 37:p. 6]

The Gartner Group attributed 56% of the total costs, by far the most significant portion, to labor related to "end-user operations." End-user costs are incurred because there may be fewer formal IS billets, and so the functions that were formerly being performed by the technicians are now performed by end-users at the departmental level. According to a noted expert, "As a distributed system evolves, users may assume a role in its design and operation. As desirable as this is, it also can result in a substantial expenditure of bootlegged time that is not explicitly identified as a development cost" [Ref. 39:p. 15]. Thus, IS personnel costs on paper may appear to be diminishing, when in reality, the costs are merely being shifted to the end-users. End-user operational costs include the following:

- Data management. This refers to the end-user requirement to perform a large degree of data manipulation formerly done by IS personnel—such as backup, conversion, compression/decompression, transfer, encryption, organization, and sharing of files.
- Application development. Ranging from spreadsheet work to coding using lower level languages, more and more informal application development is being done by the end-user to assist in job-related tasks.
- Formal learning. Classroom training time is often needed to get familiar with either applications or development tools.
- Casual learning. On-the-job learning substitutes for formal learning when formal training is not available.
- Peer support. The office PC guru spends much of his or her time substituting as the "help desk," taking that individual away from primary duties.
- Supplies. The purchase of expendable materials at the end-user level to include everything from paper to diskettes.
- "Futz" factor. The time squandered unproductively in an application, whether it is playing with different fonts or conducting personal business. [Ref. 37:p. 6-7]

b. Hidden Overhead Costs for Downsized Systems

In general, mainframe-based systems can be classified as capital intensive whereas PC LANs can be labelled as more labor intensive. In a centralized architecture, all major functions can be performed at the data center; core teams of highly specialized and trained personnel are easily capable of maintaining applications and system functionality for thousands of users. A distributed environment shifts many of those centralized functions to the department and end-user level. According to studies, some distributed systems (client/server) will require a support person for every 35 users, with support costs being three times as much as the hardware and software [Ref. 40:p. 28]. Table 6 reflects the relationship between labor and capital cost as the computing platform changes.

Table 6: LABOR & CAPITAL COSTS OF PLATFORMS [Ref. 41:p. 8]

Platform	Labor	Capital
Mainframe	30%	70%
Minicomputer	40%	60%
PC (Enterprise Network)	61%	39%
PC (LAN Work Group)	69%	31%
PC (Stand-alone)	85%	15%

As suggested by the full life-cycle cost distribution in Figure 14, distributed environments sometimes come with hidden price tags. The Gartner Group categorizes many of these hidden elements as management and control functions such as availability, performance, backup and recovery, security, system management, network management, disaster recovery, capacity planning, and change management [Ref. 41:p. 8]. It may be the case that there is no need for some of these requirements, and thus, the hidden overhead may not be as damaging. If, however, these hidden costs do represent essential functions of an organization's infrastructure *and* they are not adequately addressed in the cost estimates, a distorted picture of the true costs can result (Figure 15).

3. Intangible Benefits

Intangible benefits refer to those attributes of desktop systems that may be extremely beneficial, but hard to express in monetary terms. In fact, in the long run, these benefits may be a more significant driver in fueling the downsizing trend than the perception of cost savings. While corporate end-users are becoming accustomed to the new desktop

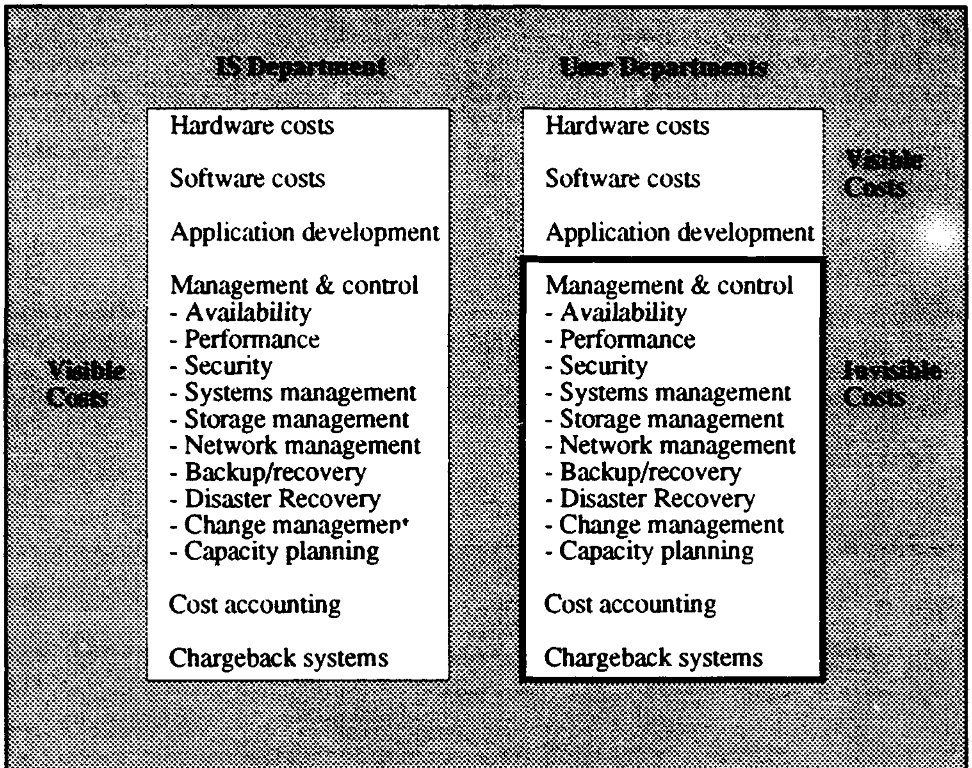


Figure 15: Management and Control Costs [Ref. 41:p. 8]

control and ease-of-use of these new desktop tools, corporate officials increasingly realize the strategic necessity of greater technological flexibility and responsiveness to change. Yet how does one measure the benefits or costs of attributes like desktop control, technological flexibility, and greater responsiveness to new business requirements? Undeniably, these factors are perceived to be critical to many organizations—but how critical, and what is it worth? The fact remains that measuring any increase or decrease in productivity as a result of these intangible benefits is a very difficult task.

Some believe that downsizing is a revolutionary process that cannot be cost-justified with traditional IS cost-estimating procedures. They argue that "traditional IS costing methods are based on the rules of a bygone era in which mass production manufacturing served as the model for cost justification" and that applying old methodologies to try to cost-justify downsizing is simply inappropriate [Ref. 42]. Others agree that downsizing is a revolutionary process, but that a cost/benefit analysis becomes that much more significant *because* of the importance of the intangible and unquantifiable costs and benefits. According to the Gartner Group:

> Downsizing is often viewed as a process that stems naturally from hardware becoming smaller, less expensive, and easier to maintain. In reality, downsizing is part of a technology restructuring that formalizes a fundamental shift of ownership of IT assets. Decisions require a fully burdened life-cycle cost analysis, especially because a substantial amount of spending occurs through end-user activities that are not part of the formal IS budget. [Ref. 38:p. 6]

Undoubtedly, cost-justifying downsizing is an extremely complex process that may require non-traditional methods. And certainly, the qualitative and largely unquantifiable factors should be weighted equally with discrete and quantitative ones. Given the limited scope of this thesis and the almost unwieldy task of a complete cost-justification, this chapter will not attempt to outline specific techniques of full-blown cost-justification—which would inevitably require weighting the indirect benefits and incorporating them with those features that are directly measurable.

V. FRAMING THE DOWNSIZING ISSUES FOR DOD AND ONI

Much of the focus of this thesis so far has been on providing a corporate perspective of downsizing issues. Earlier references to business needs and processes, organizational support, and cost savings have frequently been mentioned in the context of the civilian agencies competing in the private market. Despite the focus on the corporate world, drawing boundaries between DoD organizations and the private sector to isolate differences is largely unnecessary. DoD organizations are unique in many respects, but for the most part IT trends mirror those of its corporate counterparts.

This chapter will attempt to "frame" and summarize the key downsizing issues for DoD organizations such as the Office of Naval Intelligence (ONI). This will include an outline of the DoD policies and federal initiatives that have created a virtual *mandate* to downsize—and how DoD organizations are responding to that mandate. The last part of this chapter will highlight key issues for ONI by encapsulating "corporate lessons learned." This portion will summarize the thesis, hopefully giving top managers general guidelines with which they can intelligently plan a downsizing strategy and make key decisions.

A. THE DEPARTMENT OF DEFENSE (DOD) MANDATE

DoD and ONI are in the process of responding to the same business pressures shared by private industry. Customers are demanding better service. Budgets are declining. Subsequent pressures to "do more with less" and "work smarter, not harder" are forcing organizations to re-think the way they do business. The most common response to this challenge, both in the corporate world and government, has been to use IT to reengineer and redefine business processes and achieve greater efficiencies with tremendous cost savings. Where does the trend of downsizing information systems fit into this picture? The downsizing trend is a *manifestation* of efforts to *implement* this vision.

The federal government and DoD, responding to pressures to utilize IT more efficiently, have begun a series of initiatives to fight the complexity and costs of computing and streamline government-wide information processing.

1. Corporate Information Management Program

One of the most visible initiatives of DoD in recent years has been the Corporate Information Management (CIM) program. CIM seeks to help DoD managers and military commanders make best use of their information by calling for the restructuring and realignment of the entire information infrastructure of the Department of Defense.

The objectives of CIM sound remarkably similar to those of the business reengineering and the total quality movement (Chapter IV). CIM's goal is to optimize business processes by identifying business improvement opportunities, promoting a common understanding of business rules across departments, and then designing systems to support the way business is done. To complement and support these goals, CIM strongly advocates standardization, interoperability, shared data, reusable software, and modular applications. Examples of implementations of CIM concepts include attempts to consolidate some of DoD data centers; one phase included consolidation of 428 data centers into 52 megacenters. A more recent CIM initiative has been for DoD to consolidate 21,000 of its legacy applications into about 547 interim systems by 1999. The payoff for such bold initiatives include significant cost reductions because of fewer systems, less maintenance, shorter development times, and lower development costs. According to Paul Strassman, former director of Defense Information and a leading advocate of the CIM program:

> The savings are everywhere, because savings are [achieved], by and large, by doing business process redesign... We have a formal process for looking at business processes—who says what to whom, who passes paper to whom—and then doing value

> analysis. And [we have found] whole functions which have grown over 30 years which are not necessary anymore. [Ref. 43:p. 107]

Though CIM efforts such as data consolidation hint of centralized processing, Strassman is very much a believer in distributed architectures:

> You must also understand that we also believe in client/server. While we are consolidating the main files, at the same time we're... decentralizing application processing into a client/server environment. [Ref. 43:p. 108]

2. Defense Management Review Directive (DMRD) 918

To support the concepts defined by CIM, in early 1993 DoD issued a Defense Management Review Decision (DMRD) 918 that mandates DISA to design, develop, maintain, and support a Defense Information Infrastructure (DII) that will help DoD create a multi-service, seamless, end-to-end network based on an open system, multi-level security, client/server environment. According to the director of the Pentagon's Center for Information management, the ideal DII will create an "infosphere" that will be capable of providing military personnel with integrated services for voice, data, and video from a single terminal. Paul Strassman says DII is to DoD as the skeleton is to the body—DII will essentially serve as the backbone on which the defense databases, the CIM technical reference model, the data dictionary, and systems will hang. [Ref. 44:p. 3]

The DMRD 918 strategy for streamlining the DoD information infrastructure does not rely on merely building a conceptual framework. For example, besides working towards creating a service-wide information infrastructure, DMRD 918 also seeks to align automatic data processing (ADP) efforts through consolidation of many of independent and service-specific functions at DISA. These changes are being coordinated by new DISA director, Lt. Gen. Alonzo Short Jr., who believes "[c]hange is not going to stop. IT applications must be the directing force as you set to add value to your products and services... There has to be a transition from high volume to high value" [Ref. 45:p. 15]. As originally estimated, this new realigned infrastructure has projected savings of $4.5 billion across all the services for the period from 1993 to 1999.

3. National Performance Review (NPR)

In releasing the report of the National Performance Review (NPR) in September 1993, Vice President Gore helps to validate ongoing DoD efforts to do more with less by streamlining IS processes, reducing redundancy, and encouraging interoperability. Entitled *From Red Tape to Results, Creating a Government that Works Better & Costs Less*, the NPR issues a clarion call for using IT to reengineer business processes and obtain radical improvements in government efficiency:

> Washington's attempts to integrate information technology into the business of government have produced some successes but many costly failures. Many federal executives continue to overlook information technology's strategic role in reengineering agency practices. Agency information resource management plans aren't integrated, and their managers often aren't brought into the top realm of agency decision-making. Modernization programs tend to degenerate into loose collections of independent systems solving unique problems. Or they simply automate, instead of improve, how we do business. [Ref. 46]

The report suggests extensive use of IT to create a government that works better and costs less:

> With computers and telecommunications, we need not do things as we have in the past. We can design a customer-driven electronic government that operates in ways that, 10 years ago, the most visionary planner could not have imagined... [Ref. 46]

> As everyone knows, the computer revolution allows us to do things faster and more cheaply than we ever have before. Savings due to consolidation and modernization of the information infrastructure amount to $5.4 billion over 5 years. [Ref. 46]

B. RESPONDING TO THE MANDATE

Undoubtedly, the federal initiatives that are in place are a clear edict for the use of IT to "reinvent" government and make it do more for less money. Though the problems of government are well identified and the vision of the future is clear, the procedures for achieving these objectives are vague. Promises of a "customer-driven electronic government" and DoD "infospheres" all too often paint a rosy future without adequate discussion of procedural implementation.

Ultimately however, in terms of IT, the implications of these federal initiatives are much the same as in the corporate world. One of the inevitable responses to the mandate is to "downsize" or "rightsize" organizations, taking advantage of new and faster technologies and more flexible architectures such as client/server.

Viewed somewhat as a panacea to government ineffectiveness and inefficiency, the downsizing of information systems is already a pervasive federal trend—not necessarily in response to the federal directives. DoD and other federal agencies are off-loading their applications from the mainframes to distributed computing environments at much the same pace of the civilian sector. Of all federal IT executives surveyed in a 1993 downsizing study, 84 percent of them revealed that their organizations planned to migrate to a client/server architecture [Ref. 47:p. 16]. Such responses support other studies of declining federal mainframe usage and estimates of a continuing exodus towards desktop computing. Findings from one study regarding mainframe usage are illustrated in Figure 16.

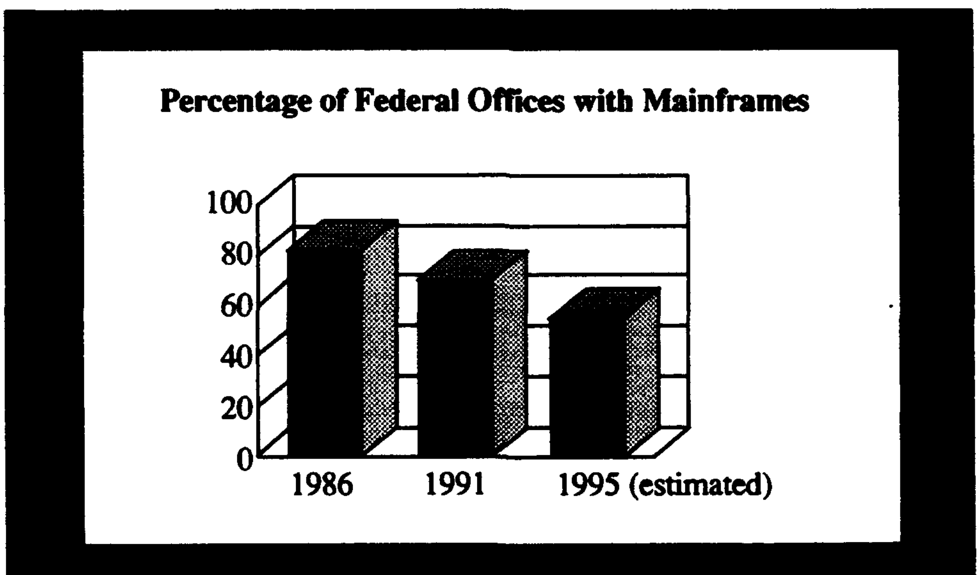


Figure 16: Mainframe's Declining Usage [Ref. 48:p. 6]

C. HIGHLIGHTING CORPORATE "LESSONS LEARNED"

In light of the exodus to smaller systems and open, distributed architectures such as client/server, downsizing of IS is too often perceived as the "easy answer" to federal problems. As the corporate world has discovered, however, while downsizing *can* resolve some issues, it also has the potential of opening up a Pandora's Box if done based on faulty organizational, technical, and cost assumptions.

At a June 1993 symposium on *Rightsizing in the Federal Government*, keynote speaker Congressman Charlie Rose (D-N.C.) warned that the growth of the distributed environment could lead to administrative nightmares for agencies. Rose said that the combination of powerful desktop systems armed with rudimentary programming knowledge could create "the new anarchy of the masses in computing." He also cautioned the audience about the functions they might be inadvertently abandoning in a downsized environment, pointing out the mainframe strengths in the area of systems and network management and security. Concerned about this possible lack of control, Rose stated that "multiple forms of bad data" and problems with viruses "could be the legacy of the new structure." As a final note, he advised "Let's not throw out all of the lessons we learned from the mainframe era." [Ref. 47:p. 16 & 20]

Thus, as ONI and other DoD agencies try to respond to pressures to "do more with less" and examine the option of downsizing information systems from a mainframe environment to distributed architectures such as client/server, strong consideration needs to be given to several key issues that have been covered in this thesis. The following paragraphs use key findings of this thesis to summarize important considerations when downsizing.

1. Organizational Considerations

- *Recognize organizational implications of distributed computing before downsizing*. Distribution of computing resources may be inherently more suitable in some organizations than others depending on organizational operations and enterprise-wide interdependencies. Although technical considerations are critical, they should not be the prime force behind a system architecture. Additionally, end-users should be given special consideration and a voice in the distribution and utilization of computing resources.

- *Understanding of business needs is critical.* IT can only be a strategic asset to the degree that it supports the corporate strategy. For downsizing to work, planning efforts must ensure that the downsized product is relevant to the business needs. Analysis of critical success factors in strategic IS planning may serve as an essential first step.
- *Optimize business processes before automation.* All aspects of a system must be broken down to their individual components to the point where all linkages, interdependencies, and relationships between parts are clearly understood. All inefficiencies in the system should be removed, allowing only the processes that add *value* to remain. Streamlining and possible automation of these remaining processes ensures maximum efficiency and productivity.
- *Top management support is a prerequisite.* Before proceeding too quickly with any downsizing plans, it is an absolute necessity to ensure that top corporate officials have "bought off" on the idea. Downsizing is as much a political issue as a technical issue. Commitment is crucial in light of a potentially difficult migration process full of volatile cultural and social upheaval.
- *Resistance to change should be anticipated and managed.* Downsizing will require change—and inevitably, change will provoke resistance. Unless proactive steps are taken to counter potentially negative reactions, the forces of resistance can work to undermine any downsizing initiatives. Thus, the CIO and CEO must acknowledge these predictable responses and work in concert to match each, in turn, with a countermeasure.
- *Some centralized direction is essential.* Considerable centralized direction from top IS management is critical to guide the downsizing process; this is particularly true with respect to the IS infrastructure.

2. Architectural and Technical Considerations

- *Client/server architecture promises the most flexibility.* In the client/server model, a network composed of mainframes (to include mid-range computers) and powerful desktop computers can all play critical roles. The varying degrees of client/server different models and "shades" of client/server computing allow data presentation, application location, and data management to be hosted on the most appropriate platform. Performance can be optimized by the shifting and redeployment of data resources and computing power according to task requirements and system strengths.
- *Heterogeneous client/server architectures also promise complexity.* The inevitable by-product of technological flexibility has been complexity. The trend has moved computing from an era of manageable, homogeneous computing on monolithic systems to an era of unmanageable, heterogeneous, networked systems creating an architectural crisis of sorts.
- *The mainframe can play critical roles in a client/server environment.* Many IS experts believe that the mainframe's role will evolve this decade in new ways that are

synergistic with the boom in distributed computing. New technologies can supplement rather than displace older ones with the mainframe being particularly well-suited as an enterprise hub in the areas of data management and networking management.

- *Open systems remain to be one of the biggest challenges*. Building a comprehensive and absolute set of standards that will allow construction of integrated systems and will allow interoperability of systems and portability of applications is unrealistic. Gartner Group predicts a coexistence of multiple standards with interoperability between them.

- *The role of middleware is inextricably linked to open systems*. Middleware, with its goal of transparent software services, will continue to be a critical component in reaching any virtually homogeneous architecture.

- *Rapid Application Development tools (RAD) can significantly increase productivity*. RAD tools such as 4GLs can offer significant benefits to increase application development productivity. Though some believe that 4GL's are too proprietary, do not permit portability, and are unsuitable for complex programs requiring intricate data handling, some 4GL tools have proven otherwise.

- *Selecting an appropriate downsizing candidate is crucial to project success*. Many IS personnel use heuristics such as high maintenance costs and low mission-criticality to guide their decisions. Other factors such as the actual structure of the code, however, can significantly affect downsizing feasibility as well.

- *Migration strategy is as important as the downsizing decision itself*. Organizations need to decide to either (1) grow with their traditional mainframes, (2) fade out the mainframe while preparing replacement systems, or (3) kill the mainframe as quickly as possible—commitment to a global strategy should help guide other decisions.

- *Downsizing can mean trading systems maturity and integrated services of larger systems for flexibility, openness, and unintegrated services of distributed systems*. Despite their proprietary nature and expense, many corporations are unwilling to move mission-critical systems to an unproven, immature desktop environment. The services provided by mainframe systems management tools are perceived to be more developed and tested than in a client/server environment.

- *Security issues should not be an afterthought*. Neglecting such important issues as security and integrity can result in serious loopholes in the information system. Despite rapid advances, the perception is that mainframe-strength security systems are simply not as available—or as robust—in a distributed environment

- *Throughput capabilities remain a major strength of the mainframe*. The mainframe has been optimized to support high volume and intricate data management with its enormous bandwidth and the ability to manage multiple complex tasks. The most advanced and developed desktop systems are just beginning to compete with the mainframe in this area.

3. Cost Considerations

- *Cost/benefit analysis must include conversion costs and costs of operating in new environment*. Often the costs associated with conversion are overshadowed by promises of cost savings in a new environment. Up-front conversion costs to distributed systems can represent a sizable investment.

- *Conversion costs will vary tremendously depending on the candidate application for downsizing and the migration strategy.* If the corporate migration strategy is to "kill" the mainframe and to off-load all applications, the conversion costs may be dramatically higher than with a "surround and integrate" strategy.
- *Costs of operating in a distributed environment are often grossly underestimated.* Focus on the desktop capital-related hardware and software costs to the exclusion of personnel and systems management costs can result in a distorted cost analysis.
- *Contrasting costs of mainframes to desktop systems in terms of price-to-performance rations can be a very misleading comparison.* The processing power does not take into account the true meaning of MIPS, nor does it account for a myriad of the other technical and non-technical issues that comprise a significant share of an IS budget.
- *In terms of costs, mainframes tend to be capital intensive; desktop systems tend to be labor intensive.* Gartner Group estimates that the capital-related costs of the average five-year life cycle of an average PC in a corporate environment accounts for *only 15%* of the total costs; the other 85% percent of the costs are personnel costs, related to administrative and technical support as well as end-user operations.
- *Cost-justifying downsizing is an extremely complex process that may require non-traditional methodologies.* Because many intangible factors such as "ease-of use" and flexibility can yield significant benefits, the qualitative and largely unquantifiable factors should be weighted equally with discrete and quantitative ones in a full-blown cost/benefit analysis.

VI. SUMMARY

This thesis has been written for middle and top managers at ONI and has sought to provide an executive-level summary of the issues involved in downsizing information systems. Downsizing of information systems is an extremely complex process that requires an understanding of a wide variety of managerial and technical issues. Baffling terminology, biased vendor assistance, and rapidly changing technology complicate already difficult decisions. This paper was conceived and written with the intention of assisting decision-makers by untangling and outlining some of those key and pertinent issues.

The traditional component that has dominated the centralized corporate IS architectures over the past 20 years has been the mainframe. With the advent of the personal computers and their rapidly improving capabilities, a paradigm shift has occurred that has enabled the desktop computers to evolve into a critically important role in current-generation IS architectures. This paradigm shift to smaller systems—downsizing—has led to a migration of IS functions off the centralized mainframe to an environment of distributed computing. Organizations and users perceive many of the desktop advantages such as cost savings, flexibility, and desktop control to be among other factors that make desktop computing a strategic necessity.

One architecture that is taking center stage as part of corporate IS is the client/server model. Within this architecture, the degree to which functions are shifted from a central computer (the server) to a desktop machine (client) varies greatly among applications. The mainframe is likely to remain an important component within this new computing paradigm. For some organizations, the mainframe is needed to perform in new capacities as an enterprise hub—both as a data server and systems manager. Because the sophistication of computing tools has advanced in parallel with new computing concepts, the capabilities of architectural tools are also fueling the downsizing movement in some cases. Though contemporary desktop tools can offer much promise, they may not necessarily be mature enough for sophisticated applications. The choice of migration strategies—for selecting the best application to downsize and transitioning to the new architecture—is one of the critical issues facing IS management.

The heart of this thesis is the risk assessment in the fourth chapter. The risks associated with downsizing were categorized according to organizational, technical, and economic factors. This chapter looked at some organizational prerequisites for downsizing including the need for understanding business needs, processes, and generating organizational support. An assessment of the technical factors revealed the need for some inevitable trade-offs when moving from mainframes to desktop systems. The cost/benefit analysis section helped to highlight some of the often hidden costs of distributed computing.

The fifth chapter "framed" the *downsizing issues for* DoD and ONI. Specifically, DoD and federal IT initiatives were outlined with the intention of showing how various programs are pressuring DoD for greater efficiency and effectiveness through IS—in effect, creating a "virtual mandate" to downsize to distributed environments. The final part of this chapter outlined the key corporate "lessons learned" and findings that decision-makers may use as a checklist to review the appropriateness of downsizing decisions.

What this thesis did *not* do was provide an in-depth case study relating many of these issues to an actual organization's downsizing experiences. As research progressed on this thesis, it became increasingly obvious that to do so would be at the expense of critical analysis of many of the core concepts. A follow-on thesis that uses the findings of this thesis as a basis for analysis of a downsizing project is recommended.

Again, the goal of this paper was to provide a "state of the art" report and risk assessment factors associated with, and affecting, the key decision to "downsize" information systems. Because there is neither a structured methodology nor "one right way" to downsize, the downsizing process can be unstructured and unwieldy. Hopefully, this broad survey has enabled decision-makers to better understand the implications of this trend by "framing" managerial and technical considerations.

LIST OF REFERENCES

1. McNurlin, B.C., and Sprague, R.H., *Information Systems Management in Practice*, 3rd ed., Prentice Hall, 1993.

2. Davidow, W.H., and Malone, M.S., *The Virtual Corporation*, HarperCollins, 1992.

3. Hammer, M., "Making the Quantum Leap," *Beyond Computing*, March/April 1992.

4. Trimmer, D., *Downsizing: Strategies for Success in the Modern Computer World*, Addison-Wesley, 1993.

5. International Business Machines (IBM), *IBM Product Pricing System/390 Announcements*, 5 September 1990.

6. Guengerich, S., *Downsizing Information Systems*, SAMS—Prentice-Hall, 1992.

7. Gartner Group Strategic Analysis Report, *From Traditional Mainframes to the New Synthesis*, by W. Malik, R.Schulte, and E. Younker, 28 July 1993.

8. Gartner Group Strategic Analysis Report, *Workstations: New Competitors for Enterprise Desktop Computing*, by G. Weiss and A. Rosenbloom, 16 August 1993.

9. Natraj, N.D., "Downsizing Information Systems," presentation (with notes collated at IA Corporation in Alameda, California) given at the Naval Postgraduate School, Monterey, California, 7 September 1993.

10. Ryan, H., "Sticker Shock! The Up-front Costs of Client/Server Can Weigh Down the Bottom Line," *Computerworld: Client/Server Journal*, November 1993.

11. Information Technology Group, *White Paper: Why Mainframes? Creating New Values for Information Systems*, 1993.

12. Gartner Group, Inc., *Inside Gartner Group this Week: Downsizing Strategies for the 1990s: Part 1*, v. 9, no. 24,16 June 1993.

13. Gartner Group, Inc., *Inside Gartner Group this Week: Downsizing Strategies for the 1990s: Part 3*, v. 9, no. 26, 30 June 1993.

14. Gerard, A., "The Client/Server Paradigm," *Data Management Review*, August 1992.

15. Van Brussel, C. "Starting Over at 55; McCormack & Dodge Founder Sees Big Future in Client/Server," *Computing Canada*, 14 September 1992.

16. Atre, S., "Client/Server and Downsizing: How to Succeed," paper presented at the Downsizing Conference, Santa Clara, California, 5 August 1993.

17. Sullivan-Trainor, M., "You're Further Along Than You Think," *Computerworld Client/Server Journal*, 11 August 1993.

18. Lash, A. "Mainframes Rise in Network Skyline," *Corporate Computing*, June/July 1992.

19. "What IS Needs to Accomplish Client/Server: the Tools, the Solutions," (advertising supplement) *Datamation*, 15 December 1992.

20. Gartner Group Research Kit, *Developing Open Systems*, by P. Magrassi, 19 April 1993.

21. DeBoevre, L., "Defining IS Architectures in the '90s: 'Shoot the Mainframe, Downsizing Triage, and Adaptive Systems,'" presentation at the Downsizing Conference, Santa Clara, California, 3 August 1993.

22. Gartner Group Research Note, *To "C" or Not to "C," or Will a 4GL Be the Answer?*, by D. Brown, 24 August 1992.

23. Gartner Group, Inc., *Inside Gartner Group this Week: Application Development: Succeeding Amid Chaos*, v. 9, no. 24,16 June 1993.

24. Defense Information Systems Agency Report, *Client/Server Guidelines to the Defense Information Technology Services Organization*, by J. Bennis, November 1992.

25. Whitten, J.L., Bentley, L.D., and Barlow, V.M., *Systems Analysis and Design Methods*, 2d ed., Irwin College Publishing, 1989.

26. Martin, L.K., *Strategic Information Planning Methodologies*, 2d ed., Prentice Hall, 1989.

27. Hammer, M. and Champy, J., *Reengineering the Corporation: A Manifesto for Business Revolution*, HarperCollins, 1993.

28. Senge, P., *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday, 1990.

29. Bolman, L.G., and Deal, T.E., *Reframing Organizations: Artistry, Choice, and Leadership*, Jossey-Bass Publishers, 1991.

30. Hallet, M., "Fault Tolerance May Be the Answer," *Computing Canada*, v. 19, 12 October 1993.

31. Gartner Group, Inc., "Open Systems Reader Notes," published notes normally provided as part of a conference presentation, 1993.

32. Baker, R.H., *Downsizing: How to Get Big Gains from Smaller Systems*, McGraw-Hill, Inc., 1992.

33. McPartlin, J.P., "Teaching Old Pros New Tricks," *Information Week*, 28 June 1993.

34. Lorin, H., *Aspects of Distributed Computing Systems*, 2d ed., John Wiley & Sons, Inc., 1988.

35. Ricciuti, M., "PCs Break Into the OLTP Ranks," *Datamation*, 15 August 1993.

36. Harding, E., "Downsizing Raises Some Risks; Companies Often Overlook Security Concerns," *Software Magazine*, October 1993.

37. Gartner Group, Inc., *Inside Gartner Group this Week: Downsizing Strategies for the 1990s: Part 4*, v. 9, no. 24, 16 June 1993.

38. Gartner Group, Inc., *Inside Gartner Group this Week: Downsizing Strategies for the 1990s: Part 6*, v. 9, no. 29, 21 July 1993.

39. Emery, J., *Managerial and Economic Issues in Distributed Computing*, Auerbach Publishers, Inc., 1978.

40. Ambrosio, J., "Client/Server Costs More Than Expected," *Computerworld*, 18 October 1993.

41. Gartner Group, Inc., *Inside Gartner Group this Week: Downsizing Strategies for the 1990s: Part 5*, v. 9, no. 28, 14 July 1993.

42. Eskow, D., "The 'D' Word," *Corporate Computing*, November 1993.

43. Interview between P. Strassman, Director, Defense Information Systems Agency, and C. Staiti with P. Pinella, senior and managing editor of *Datamation*, 15 October 1992.

44. Endoso, J., "DoD Plans its 'Infosphere'; Model of Central Information Network Coming Soon," *Government Computer News*, 18 January 1993.

45. Power, K., "Data Center Execs Receive the Word: High Value, Not Volume," *Government Computer News*, 6 December 1993.

46. White House, *National Performance Review*, 7 September 1993.

47. Moore, J., "Congress Urges Cautious Approach to Distributed Computing," *Federal Computer Week*, 21 June 1993.

48. Corbin, L., "Federal Computing," *Government Executive*, March 1992.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6415	2
2.	Library Code 52 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Mr. Brent Snyder Head, Telecommunications Management Department Office of Naval Intelligence (ONI) – 62 4600 Silver Hill Road Washington, D.C. 20389-5000	2
4.	CDR Douglas J. Beauvais Head, Fleet Systems Department Office of Naval Intelligence (ONI) – 63 4600 Silver Hill Road Washington, D.C. 20389-5000	1
5.	Mr. Donald Gerbozy Fleet Support Office of Naval Intelligence (ONI) – 734 4600 Silver Hill Road Washington, D.C. 20389-5000	1
6.	Dr. Michael J. Mestrovich Director, Office of Integration and Interoperability 5201 Leesburg 3 Skyline Place Suite 1501 Falls Church, Virginia 22041-3201	1
7.	Dr. James C. Emery Naval Postgraduate School Code SM/Ey Monterey, California 93943-5000	2

8. Mr. Nat D. Natraj 1
IA Corporation
1301 Harbor Bay Parkway
Alameda, California 94501

9. Dr. Myung W. Suh 1
Naval Postgraduate School
Code SM/Su
Monterey, California 93943-5000

10. LT Peter M. Hutson 2
1800 Beckley Drive
Chesapeake, Virginia 23320